



MII EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER

DISEÑO DE UN MODELO DE APRENDIZAJE PROFUNDO PARA REALIZAR OPERACIONES DE COMPRAVENTA DE ACTIVOS FINANCIEROS

Autor: Lucía Díaz García

Directores: Álvaro López López

Lucía Güitta López

Madrid

Julio de 2020

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
DISEÑO DE UN MODELO DE APRENDIZAJE PROFUNDO PARA
REALIZAR OPERACIONES DE COMPRAVENTA DE ACTIVOS
FINANCIEROS

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 2019/2020 es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos. El Proyecto no es
plagio de otro, ni total ni parcialmente y la información que ha sido tomada
de otros documentos está debidamente referenciada.

Fdo.: Lucía Díaz García

Fecha: 20/ 07/ 2020



Autorizada la entrega del proyecto

DIRECTORES DEL PROYECTO



Fdo.: Lucía Güitta López

Fecha: 20 / 07 / 20

Fdo.: Álvaro Jesús López López

Fecha: 20 / 07 / 20



MII EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER

DISEÑO DE UN MODELO DE APRENDIZAJE PROFUNDO PARA REALIZAR OPERACIONES DE COMPRAVENTA DE ACTIVOS FINANCIEROS

Autor: Lucía Díaz García

Directores: Álvaro López López

Lucía Güitta López

Madrid

Julio de 2020

DISEÑO DE UN MODELO DE APRENDIZAJE PROFUNDO PARA REALIZAR OPERACIONES DE COMPRAVENTA DE ACTIVOS FINANCIEROS

Autora: Díaz García, Lucía

Directores: López López, Álvaro

Güita López, Lucía

Entidad colaboradora: ICAI –Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

En este proyecto se explora una de las áreas en las que la inteligencia artificial ha generado especial interés en los últimos años, los mercados financieros. En concreto, está centrado en lo que se conoce como *trading* inteligente, es decir, la identificación de oportunidades de compraventa de activos financieros haciendo uso de técnicas propias de la inteligencia artificial. Las técnicas y herramientas elegidas para llevar a cabo este trabajo se basan en dos principios.

Por un lado, en el hecho de que los agentes de inversión, a la hora de tomar decisiones acerca de las acciones óptimas a realizar, ya sean compras o ventas de activos, incluso la decisión de no realizar ninguna operación, ejecutan sus órdenes mientras observan en una pantalla imágenes de series temporales financieras. Esta labor visual suele ir acompañada de un análisis numérico, basado principalmente en la aplicación de algoritmos, y resulta esencial al estar apoyada de la experiencia e intuición de los inversores profesionales.

En segundo lugar, gracias a los grandes avances en el rendimiento de modelos de aprendizaje profundo, debido a la cantidad creciente de datos disponibles así como la capacidad computacional, el reconocimiento de imágenes ha logrado alcanzar un gran éxito en los últimos años.

Por estas dos razones, las técnicas de inteligencia artificial empleadas para realizar este proyecto están centradas en el área de la visión artificial. Ya que resulta natural tratar de aproximar la forma de abordar un problema de inteligencia artificial a la inteligencia humana, en este caso que los inversores que cuentan con amplia experiencia realizan esta tarea de

compraventa de activos de forma visual, tiene sentido estudiar el valor de transformar el enfoque clásico de análisis de series temporales financieras numéricas en un problema de clasificación de imágenes.

Este trabajo está basado en el artículo *Trading via Image Classification* de JP Morgan. En su estudio, analizan los datos de todas las acciones del índice S&P 500 del periodo de 2010 a 2017. Comparan tres indicadores binarios diferentes conocidos, correspondiendo a cada uno una serie de normas algebraicas que determinan los diferentes momentos en los que habría una oportunidad compra, basándose únicamente en el valor de cierre diario de las acciones. Estos algoritmos tienen en cuenta los datos correspondientes a una ventana de tiempo de entre 20 y 27 días, dependiendo de la fórmula utilizada en cada uno. Por ello, las gráficas generadas incluyen la representación del día para el que se calcula el resultado del algoritmo y los correspondientes días anteriores que influyen en ese resultado. Las imágenes son etiquetadas en función de si en el último día representado hay una señal de compra o no.

Con las gráficas generadas, entrenan 16 modelos de aprendizaje supervisado diferentes con el objetivo de estudiar el rendimiento global de todos los modelos a la hora de realizar la tarea de clasificación cuando las entradas son imágenes, no siendo el objetivo elegir el mejor modelo.

Para evaluar el rendimiento de los modelos una vez entrenados, para cada día del año 2018, generan una imagen con las curvas correspondientes a los datos de los valores diarios de las acciones de dicho día y de los días anteriores relevantes, igual que para las imágenes del entrenamiento. Para cada imagen predicen si a ese día correspondería una señal de compra o no. La precisión global obtenida a la hora de detectar las señales es superior al 80% para los tres algoritmos, y superior al 90% para dos de ellos.

Se puede concluir a partir de estos resultados, que la transformación del análisis de series temporales en un problema de visión resulta beneficioso a la hora de trabajar con datos financieros para identificar oportunidades de compra de activos. Partiendo de los conceptos y métodos descritos en este apartado, y del éxito demostrado, este trabajo se divide en cuatro objetivos principales.

En primer lugar, se buscará replicar los resultados del estudio. En concreto, por su utilidad para resolver problemas de visión artificial, se utilizarán modelos de redes neuronales convolucionales de entre todos los analizados en el artículo. Las imágenes que recibirá el modelo serán las correspondientes a la evolución del valor de las acciones de varias de las

empresas del IBEX 35, en el periodo de tiempo de 2010 a 2017. Se evaluará la precisión con los datos de las mismas empresas para el año 2018. El algoritmo financiero escogido para la toma de decisiones de compraventa es el de las Bandas de Bollinger. Este algoritmo de compone de tres curvas, calculadas a partir del precio de cierre de cada acción. La primera curva es una media móvil, calculada para una ventana de tiempo de 20 días. A la media móvil, se suma y resta dos desviaciones estándar (calculadas para los mismos 20 días), calculando así las dos Bandas de Bollinger, una superior y una inferior. El corte de la curva del precio con la curva superior representará una oportunidad de venta, y con la curva inferior, de compra.

En este trabajo, a diferencia del estudio realizado por los expertos de JP Morgan, se trabajará tanto con las oportunidades de compra así como las de venta, siguiendo los mismos principios. En la Ilustración 1, se muestra un ejemplo de imagen que recibirá el modelo convolucional, en este caso correspondiente a una imagen con señal de venta. Las imágenes incluyen la representación de la curva de la evolución del precio de cierre de la acción durante 20 días, así como las curvas correspondientes al algoritmo de Bollinger, que determinarán la acción óptima a realizar el día 20.

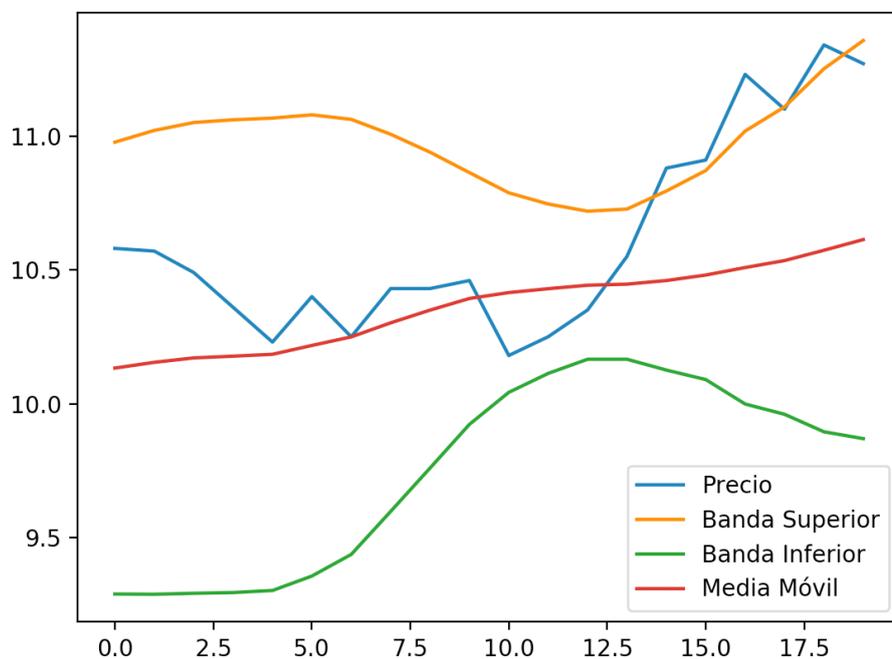


Ilustración 1. Ejemplo de imagen de entrada al modelo neuronal convolucional

Una vez conseguidos unos resultados similares, se realizarán diferentes ampliaciones al estudio, añadiendo en cada una de ellas un punto de complejidad para conseguir una aplicación más práctica, así como comprobar la eficacia y versatilidad de estos modelos para distintas tareas y configuraciones de los datos. Se realizarán dos cambios diferentes en los datos de entrada para evaluar el rendimiento del modelo en casos en los que se aporte menor información. En primer lugar, de las gráficas iniciales se eliminará el último día representado, al que corresponde la acción a tomar y por la cual ha sido clasificada la imagen. De esta forma, se realizará una tarea de predicción. El modelo, conociendo los datos de los 19 días anteriores, deberá predecir la acción a realizar el día siguiente. En la Ilustración 2 se distinguen las imágenes que reciben el primer modelo, con información de 20 días, y el segundo modelo de clasificación, cuyas imágenes incluyen la representación de 19 días. En este caso, el modelo deberá reconocer para la imagen de la derecha, que al día siguiente habría una oportunidad de compra.

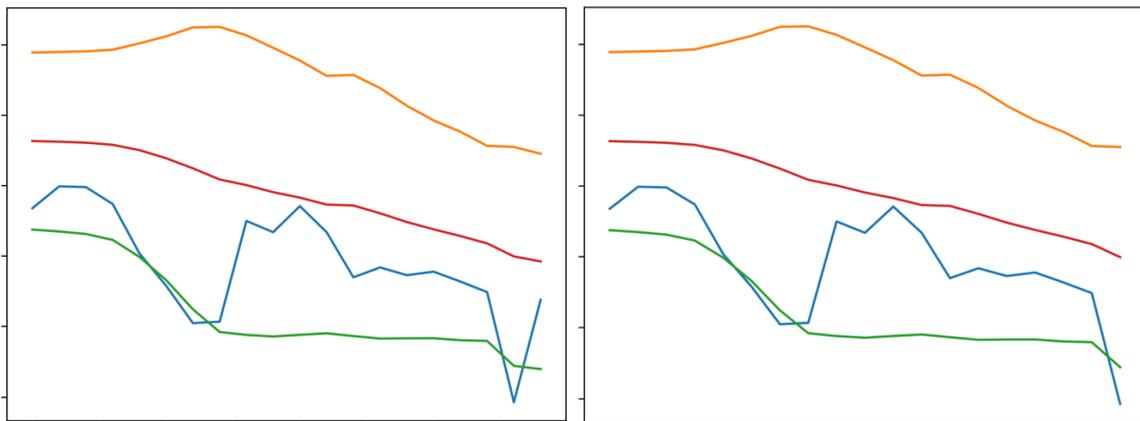


Ilustración 2. Ejemplo de imagen de entrada al modelo convolucional de predicción

La segunda variación consistirá en eliminar de las gráficas las curvas correspondientes al algoritmo de las Bandas de Bollinger. En este caso se buscará que el modelo aprenda solo el algoritmo y sea capaz de clasificar las imágenes correctamente conociendo únicamente el valor del precio de cierre de las acciones. En la Ilustración 3 se puede ver un ejemplo del resultado de eliminar las curvas del algoritmo de las imágenes.

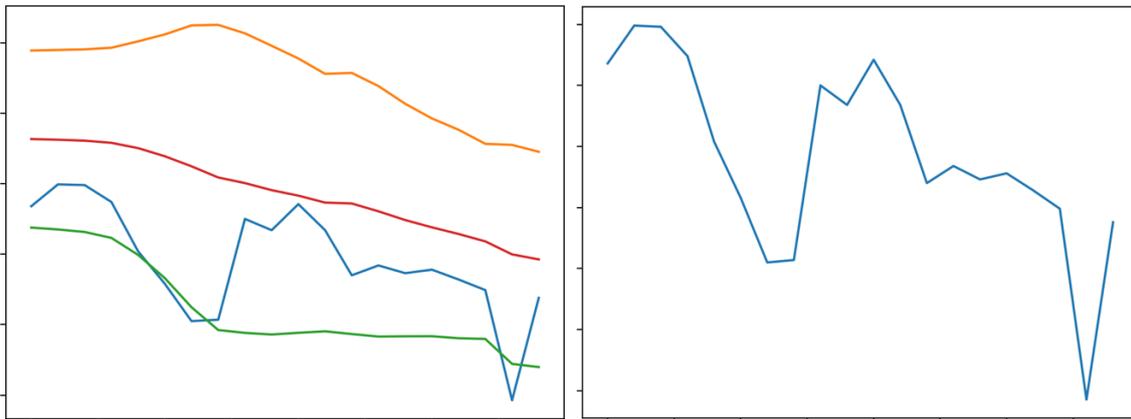


Ilustración 3. Ejemplo de imagen eliminando la representación de las curvas del algoritmo de Bollinger

En tercer lugar, se entrenará un modelo convolucional, que recibirá las mismas imágenes que el primer modelo, para realizar una tarea de regresión, con el objetivo de identificar el punto exacto de corte de la curva del precio con una de las Bandas de Bollinger.

Por último, se realizará una comparación entre los resultados obtenidos con los modelos convolucionales, siendo los datos de entrada imágenes, y los resultados obtenidos con modelos de machine learning clásicos, que toman como entrada datos numéricos en forma tabular. El objetivo de esta comparación es evaluar la utilidad y viabilidad de convertir este problema financiero de análisis temporal en uno de visión artificial.

En la Tabla 1 se recogen los resultados de los modelos convolucionales de clasificación y predicción. La precisión se ha calculado como el porcentaje de las muestras del total correctamente clasificadas. Como se puede observar, tres de los cuatro modelos alcanzan una precisión cercana al 85%. El modelo convolucional que realiza una predicción conociendo únicamente la curva del precio es el modelo para el que se ha obtenido la mayor precisión, del 85,5%. Este modelo es el más interesante ya que al ser el que trabaja con menor información es el más útil y práctico.

MODELO	PRECISIÓN
Clasificación, representación de todas la curvas	84,5%
Clasificación, únicamente curva del precio	85,3%
Predicción, representación de todas las curvas	53,5%
Predicción, únicamente la curva del precio	85,5%

Tabla 1. Resultados de precisión finales de todos los modelos convolucionales

Se han entrenado tres modelos diferentes de machine learning, una regresión lineal, una máquina vector de soporte y un random forest. Estos modelos toman como datos de entrada un vector de 20 posiciones, con la información del valor del precio de cierre de cada acción durante 20 días. Estos modelos son entrenados con la misma información que el modelo neuronal convolucional de clasificación para el cual los datos de entrada eran imágenes con la representación del precio de cierre únicamente. La diferencia entre la información que reciben los modelos de machine learning respecto al modelo convolucional es que para los primeros el formato es numérico, mientras que para el segundo son imágenes. Por esto, los resultados de los modelos de machine learning se compararán con este modelo convolucional de todos los entrenados.

Los resultados de la precisión de estos modelos se muestran en la Tabla 2. Como se puede observar, la red neuronal convolucional presenta una precisión superior a la de todos los modelos de machine learning, siendo entre un 34,5% y un 73,3% mayor.

MODELO	PRECISIÓN
Regresión lineal	12%
Support Vector Machine	50,8%
Random forest	28,8%
Red neuronal convolucional	85,3%

Tabla 2. Comparación de los resultados de los modelos de machine learning frente a una red neuronal convolucional

De las dos tablas de resultados se pueden extraer dos conclusiones principales. En primer lugar, los modelos neuronales convolucionales son capaces de recuperar con éxito la mayor parte de las señales de compra y de venta, distinguiendo a su vez los días en los que no se presenta ninguna de estas señales. Con estos resultados, se puede confirmar que al transformar el planteamiento clásico de este problema financiero centrado en la compraventa de activos de un problema numérico a uno visual se consigue realizar la tarea de clasificación con éxito.

Por otro lado, comparando con los resultados obtenidos por los modelos de machine learning, se confirma que no sólo son los modelos neuronales convolucionales capaces de resolver este problema, si no que superan con creces los resultados de estos modelos más simples. Los modelos de machine learning demuestran no ser lo suficientemente complejos para clasificar las muestras correctamente, por lo que resulta beneficioso emplear modelos neuronales convolucionales.

DESIGN OF A DEEP LEARNING MODEL FOR CONDUCTING FINANCIAL ASSET PURCHASE AND SALE TRANSACTIONS

This project explores one of the areas in which artificial intelligence has generated special interest in recent years, namely the financial markets. Specifically, it focuses on what is known as intelligent trading, i.e., the identification of opportunities for buying and selling financial assets using artificial intelligence techniques. The techniques and tools chosen to carry out this work are based on two principles.

On the one hand, in the fact that traders, when making decisions about the optimal actions to take, whether they are buying or selling assets, or even the decision not to carry out any operation, execute their orders while observing images of financial time series on a screen. This visual work is usually accompanied by a numerical analysis, mainly based on the application of algorithms, and is essential as it relies on the experience and intuition of professional investors.

Secondly, thanks to the great advances in the performance of deep learning models, due to the increasing amount of data available as well as the computational capacity, image recognition has achieved great success in recent years.

For these two reasons, the artificial intelligence techniques used to carry out this project are focused on the area of machine vision. Since it is natural to try to approximate the way of

approaching an artificial intelligence problem to that of human intelligence and in this case investors with extensive experience perform this task of buying and selling assets visually, it makes sense to study the value of transforming the classic approach of numerical financial time series analysis into an image classification problem.

This paper is based on the article Trading via Image Classification by JP Morgan. In their study, they analyse the data for all stocks in the S&P 500 index for the period 2010 to 2017. They compare three different known binary indicators, each corresponding to a set of algebraic rules that determine the different times when there would be a buying opportunity, based only on the daily closing value of the stocks. These algorithms take into account data corresponding to a time window of between 20 and 27 days, depending on the formula used in each one. They then generate graphs which represent the day for which the result of the algorithm is calculated and the corresponding previous days which influence this result. The images are labeled according to whether or not there is a purchase signal on the last day represented.

With the generated images, they train 16 different supervised learning models with the aim of studying the overall performance of all the models when performing the classification task when the entries are images, the objective not being to choose the best model.

To evaluate the performance of the models once they have been trained, for each day of the year 2018, they generate an image with the curves corresponding to the data of the daily values of the stocks for that day and the previous relevant days, the same as for the training images. For each image they predict whether a buy signal would correspond to each day or not. The overall accuracy obtained in detecting the signals is over 80% for all three algorithms.

It can be concluded from these results that the transformation of time series analysis into a vision problem is beneficial when working with financial data to identify asset purchase opportunities. Based on the concepts and methods described, and the success demonstrated, this work is divided into four main objectives.

First, we will seek to replicate the results of the study. In particular, due to their usefulness in solving artificial vision problems, models of convolutional neural networks will be used from among all those analysed in the article. The images that the model will receive will be those corresponding to the evolution of the value of the shares of several of the IBEX 35 companies, in the period of time from 2010 to 2017. The accuracy will be evaluated with the data of the same companies for the year 2018. The financial algorithm chosen for the buying and selling

decisions is the Bollinger Bands algorithm. This algorithm is composed of three curves, calculated from the closing price of each share. The first curve is a moving average, calculated for a 20-day time window. Two standard deviations (calculated for the same 20 days) are added to and subtracted from the moving average, thus calculating the two Bollinger Bands, one upper band and one lower band. The crossing of the price curve with the upper band will represent a selling opportunity, and with the lower band, a buying opportunity.

In this work, unlike the study carried out by the experts of JP Morgan, both buying and selling opportunities will be taken into account, following the same principles. Figure 1 shows an example of an image that will receive the convolutional model is shown, in this case corresponding to an image with a selling signal. The images include the representation of the curve of the evolution of the closing price of the stock during 20 days, as well as the curves corresponding to the Bollinger algorithm, which will determine the optimal action to be carried out on the 20th day.

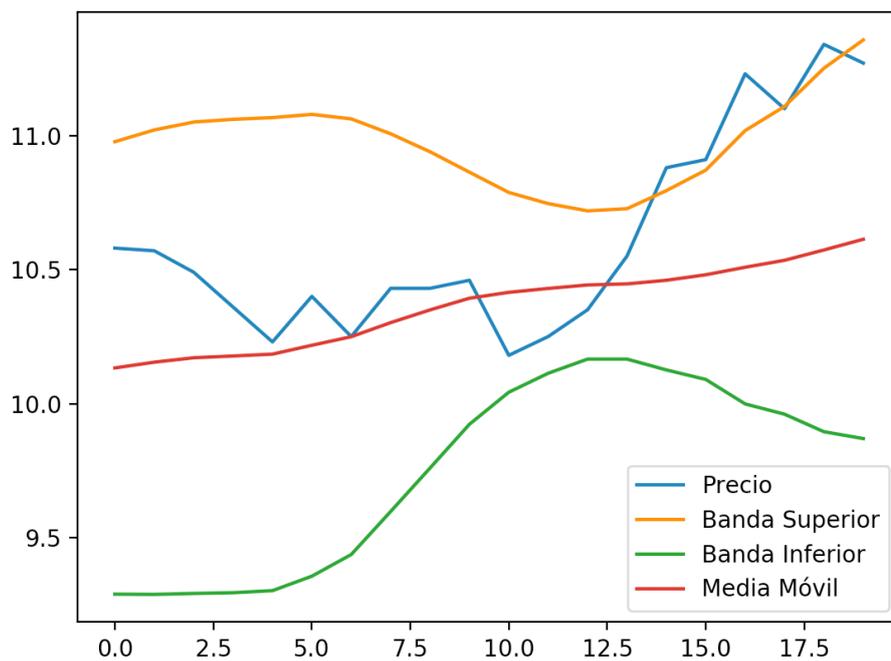


Figure 1. Example of an input image for the convolutional neural network

Once similar results have been achieved, different extensions will be made to the study, adding a point of complexity in each of them to achieve a more practical application, as well as checking the effectiveness and versatility of these models for different tasks and data configurations. Two different changes will be made to the input data to assess the performance

of the model in cases where less information is provided. First, the last day represented will be removed from the initial graphs, which is the day that corresponds to the action to be taken and for which the image has been classified. In this way, a prediction task will be performed. The model, knowing the data of the previous 19 days, must predict the action to be taken the next day. Figure 2 shows the images that receives the first model, with information for 20 days, and the second model, with images that include the representation of 19 days. In this case, the model must recognize for the image on the right, that the next day there would be a buying opportunity.

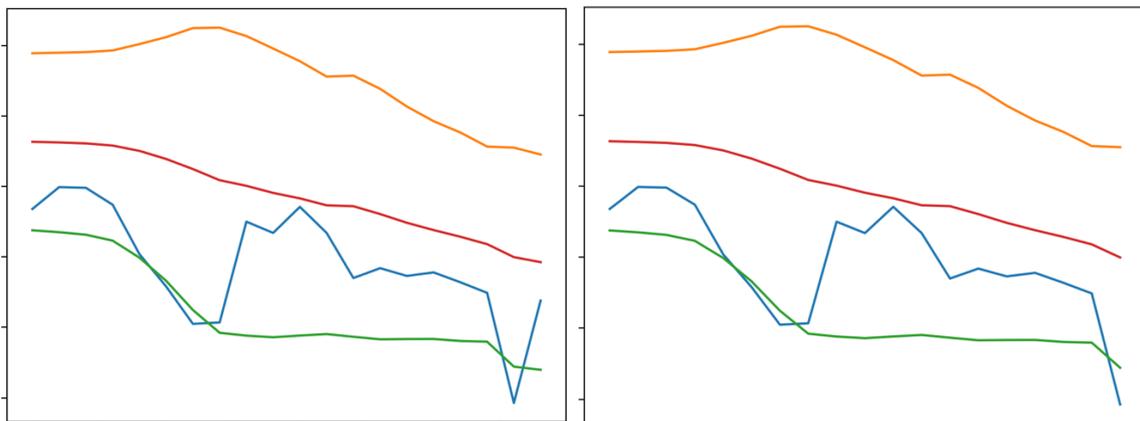


Figure 2. Example of input image for the predictive neural network

The second change will consist of removing the curves corresponding to the Bollinger Bands algorithm from the images. In this case the model should be able to learn the algorithm on its own and be able to classify the images correctly knowing only the value of the closing price of the stocks. Figure 3 shows an example of the result of removing the curves of the algorithm from the images.

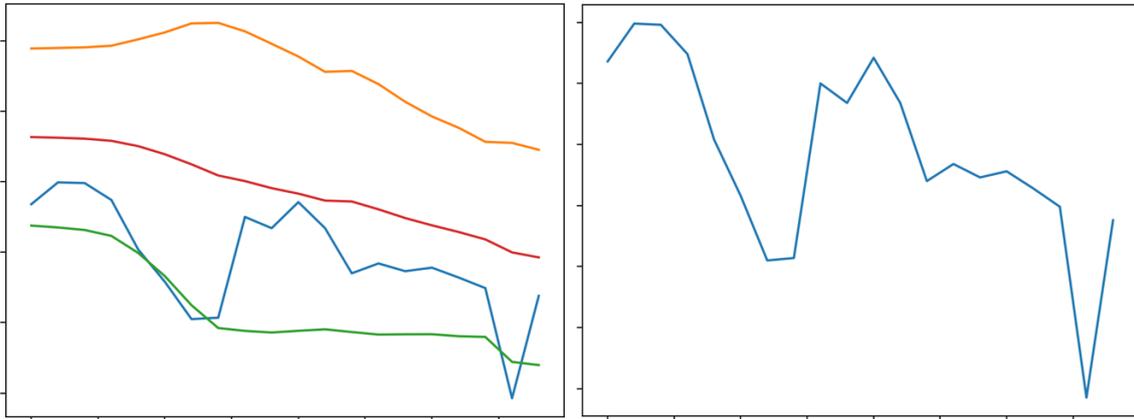


Figure 3. Example of an input image removing the curves from de Bollinger Algorithm

Thirdly, a new convolutional model will be trained, which will receive the same images as the first model, to perform a regression task, with the aim of identifying the exact crossing point of the price curve with one of the Bollinger Bands.

Finally, a comparison will be made between the results obtained with the convolutional models, being the input data images, and the results obtained with classic machine learning models, which take as input numerical data in tabular form. The aim of this comparison is to evaluate the usefulness and feasibility of converting this financial problem of time-series analysis into one of artificial vision.

The results of the convolutional classification and prediction models are shown in Table 1. The accuracy has been calculated as the percentage of the samples of the total correctly classified. As it can be seen, three of the four models achieve an accuracy close to 85%. The convolutional model that makes a prediction knowing only the information about the price curve is the model for which the highest accuracy, 85.5%, has been obtained. This model is the most interesting since it is the one that works with the least information making it the most useful and practical.

MODEL	ACCURACY
Classification, representation of all curves	84,5%
Classification, price curve only	85,3%
Prediction, representation of all curves	53,5%
Prediction, price curve only	85,5%

Table 1. Accuracy results for the convolutional neural networks

Three different models of machine learning have been trained, a linear regression, a support vector machine and a random forest. These models take as input a vector of 20 positions, with the information of the value of the closing price of each share during 20 days. These models are trained with the same information as the classification convolutional neuronal network for which the input data were images with the representation of the closing price only. The difference between the information received by the machine learning models and the convolutional model is that for the former the format is numerical, while for the latter they are images. Therefore, the results of the machine learning models will be compared with this convolutional model of all the trained ones.

The results of the accuracy of these models are shown in Table 2. As it can be seen, the convolutional neural network has a higher accuracy than all the machine learning models, being between 34.5% and 73.3% higher.

MODEL	ACCURACY
Linear Regression	12%
Support Vector Machine	50,8%
Convolutional neural network	28,8%
Red neuronal convolucional	85,3%

Table 2. Accuracy results for the machine learning models

Two main conclusions can be drawn from the two results tables. Firstly, convolutional neural models are able to successfully recover most of the buy and sell signals, while distinguishing the days when neither of these signals is present. With these results, it can be confirmed that by transforming the classical approach to this financial problem focused on the purchase and

sale of assets from a numerical problem to a visual one, the classification task can be successfully performed.

On the other hand, comparing with the results obtained by the machine learning models, it's confirmed that not only are the neuronal convolutional models capable of solving this problem, but that they far exceed the results of these simpler models. The machine learning models prove not to be complex enough to classify the samples correctly, so it is beneficial to use convolutional neural models.

Índice de la memoria

<i>Índice de figuras</i>	3
<i>Índice de tablas</i>	4
1. Introducción	5
1.1. Mercados financieros	5
1.2. Inteligencia artificial y mercados financieros	6
1.3. Objetivos del proyecto	8
2. Estado del arte	13
2.1. Aprendizaje profundo	13
2.2. Redes neuronales convolucionales	15
3. Metodología	21
3.1. Adquisición y procesado de datos	21
3.2. Modelos neuronales convolucionales	28
3.2.1. Modelo de clasificación de señales de compraventa pasadas	33
3.2.2. Modelo de predicción de señales de compraventa futuras.....	38
3.2.3. Modelo de regresión	42
3.3. Modelos de machine learning	48
3.3.1. Modelo de regresión lineal.....	49
3.3.2. Modelo de support vector machine.....	50
3.3.3. Modelo random forest.....	51
4. Análisis de los resultados	53
5. Conclusiones y futuros desarrollos	57
6. Referencias	59

UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
MÁSTER EN INGENIERÍA INDUSTRIAL

Índice de figuras

Ilustración 1. Resultados de la predicción de señales de venta de JP Morgan _____	10
Ilustración 2. Comparación del rendimiento de modelos de deep learning frente a modelos de machine learning en función del número de datos [11] _____	14
Ilustración 4. Arquitectura común de una red neuronal convolucional [13] _____	16
Ilustración 5. Ejemplo de operación de convolución [14] _____	16
Ilustración 6. Representación de la función ReLU [15] _____	18
Ilustración 7. Ejemplo de representación de las Bandas de Bollinger para la acción BBVA durante el año 2018 _____	24
Ilustración 8. Muestra de imagen correspondiente a una señal de compra _____	26
Ilustración 9. Muestra de imagen correspondiente a una señal de venta _____	26
Ilustración 10. Muestra de imagen correspondiente a un día que no presenta señal _____	27
Ilustración 11. Ejemplo de muestra incluyendo la representación de todas las curvas _____	36
Ilustración 12. Ejemplo de muestra incluyendo la representación únicamente de la curva del precio de cierre _____	37
Ilustración 13. Ejemplo de muestra que incluye la representación de 20 días para todas las curvas _____	39
Ilustración 14. Ejemplo de muestra que incluye la representación de 19 días para todas las curvas _____	40
Ilustración 15. Representación del punto de corte para una señal de venta _____	44
Ilustración 16. Comparación de la predicción del punto de corte respecto al punto real para una muestra con señal de compra _____	46
Ilustración 17. Comparación de la predicción del punto de corte respecto al punto real para una muestra con señal de venta _____	46
Ilustración 18. Comparación de la predicción del punto de corte respecto al punto real para una muestra sin señal _____	47
Ilustración 19. Ejemplo de diferentes márgenes para los mismos datos _____	50

Índice de tablas

Tabla 1. Acciones del IBEX 35 disponibles en el paquete Investpy	23
Tabla 2. Resultados de precisión para los modelos convolucionales de clasificación de señales pasadas, para imágenes con la representación de todas las curvas	34
Tabla 3. Resultados de precisión para los modelos convolucionales de clasificación de señales pasadas, para imágenes con la representación únicamente de la curva del precio de cierre	37
Tabla 4. Comparación de los resultados de la clasificación frente al valor real de las muestras.....	38
Tabla 5. Resultados de precisión para los modelos convolucionales de clasificación de señales futuras, para imágenes con la representación de todas las curvas.....	40
Tabla 6. Comparación de los resultados de la predicción frente al valor real de las muestras para las imágenes que incluyen la representación de todas las curvas.....	41
Tabla 7. Resultados de precisión para los modelos convolucionales de clasificación de señales futuras, para imágenes con la representación únicamente del precio	42
Tabla 8. Comparación de los resultados de la predicción frente al valor real de las muestras para las imágenes que incluyen la representación únicamente del precio de cierre.....	42
Tabla 9. Resultados del error cuadrático medio de los modelos convolucionales de regresión	45
Tabla 10. Resultados de precisión del modelo de regresión lineal	49
Tabla 11. Comparación de las predicciones frente a los valores reales de las muestras para el modelo de regresión lineal	49
Tabla 12. Resultados de precisión del modelo SVM	51
Tabla 13. Comparación de las predicciones frente a los valores reales de las muestras para el modelo de SVM	51
Tabla 14. Resultado de precisión del modelo Random Forest.....	52
Tabla 15. Comparación de las predicciones frente a los valores reales de las muestras para el modelo de Random Forest	52
Tabla 16. Resultados de precisión finales de todos los modelos convolucionales	54
Tabla 17. Resultados del error cuadrático medio de los modelos convolucionales de venta y compra 55	
Tabla 18. Comparación de los resultados de los modelos de machine learning frente a una red neuronal convolucional.....	55

1. Introducción

1.1. Mercados financieros

Un mercado es un lugar donde se intercambian bienes y servicios. Generalmente, los participantes se distinguen en compradores y vendedores. El tamaño de un mercado está determinado por el número de compradores y vendedores, así como la cantidad de dinero que circula durante un año. [1]

En un mercado financiero se intercambian entre otros, valores, monedas o bonos. Estos mercados pueden ser tanto físicos como virtuales, y son la base de las sociedades capitalistas, asignando recursos y creando liquidez para las empresas.

Existen diferentes tipos de mercados financieros, entre ellos [2]:

- **Mercado de valores:** en el mercado de valores se intercambian acciones de compañías públicas. Las acciones se compran a un precio determinado, y los inversores pueden ganar dinero cuando sus acciones se encuentran en buena posición en el mercado. Existen diversos índices en los cuales basarse para saber la situación del mercado, como el IBEX 35 o S&P 500
- **Mercado de bonos:** en este mercado, se pueden comprar bonos tanto de una compañía como de un estado. El vendedor debe devolver el precio del bono más un interés conocido en un periodo de tiempo predefinido. Los emisores de los bonos buscan financiar proyectos y operaciones
- **Mercado de commodities:** en el mercado de materias primas, los inversores compran y venden recursos naturales y materias primas como oro o petróleo
- **Mercado de derivados:** en este mercado se pueden encontrar derivados o contratos cuyo valor depende del valor de mercado de un activo. Un ejemplo es el mercado de futuros de materias primas, un derivado del mercado de commodities

- **Mercado Forex:** es el mercado en el cual se opera con diferentes monedas. Al ser el bien intercambiando dinero, este es el mercado con más liquidez que existe, y el que cuenta con mayor número de transacciones diarias

1.2. Inteligencia artificial y mercados financieros

Las personas siguen siendo esenciales en las operaciones de inversión, pero la inteligencia artificial cada vez tiene un papel mayor.

El mundo actual cambia a tan alta velocidad que es más indispensable que nunca ser capaz de detectar nuevas oportunidades en el mercado. La inteligencia artificial puede ser la herramienta necesaria para desarrollar estrategias financieras que previamente eran difíciles de predecir. Hoy en día muchas de las órdenes ejecutadas son realizadas por sistemas que toman decisiones basadas en algoritmos que reciben constantemente datos actualizados [3].

Más allá de utilizar algoritmos, la inteligencia artificial puede reunir datos de diferentes fuentes como periódicos, redes sociales o indicadores de intercambio para mejorar la toma de decisiones al conseguir una comprensión avanzada de las tendencias globales.

Algunos ejemplos de aplicación de la inteligencia artificial en ámbitos financieros son los siguientes [4]:

- Mejora de la gestión de riesgos y toma de decisiones: con tecnologías de IA, es posible tomar decisiones racionales no sesgadas por factores humanos. De este modo se podrán tomar las decisiones óptimas y evitar posibles errores propios de las personas.
- Mejora de la detección de fraude: la inteligencia artificial posibilita detectar patrones a partir de datos no procesados para identificar comportamientos fuera de lo normal y así detectar posibles fraudes y alertar de ello.
- Uso de “agentes virtuales”: Los agentes y asesores que proporcionan recomendaciones a los clientes que buscan realizar transacciones financieras pueden ser sustituidos por

un chatbot. Estos agentes virtuales utilizan los datos de cada cliente para proporcionar una atención personalizada.

- **Trading inteligente:** diferentes algoritmos pueden ser entrenados para crear sistemas de inversión inteligentes y conseguir beneficios consistentes. Estos sistemas pueden aprender de errores pasados, tomando así mejores decisiones en el futuro, aumentando su precisión de predicción. Estos sistemas principalmente identifican tendencias en el mercado para realizar las predicciones.
- **Transacciones de Blockchain:** la inteligencia artificial puede ser utilizada para mejorar la velocidad y el coste del blockchain. Por otro lado, la tecnología blockchain puede mejorar la seguridad y la transparencia en los sistemas de inteligencia artificial. En lugar de ser personas las que verifiquen las transacciones realizadas por blockchain, esto puede realizarse a través de un sistema inteligente, reduciendo fallos y aumentando la eficiencia.

Son numerosas las compañías que hacen uso de la inteligencia artificial para mejorar sus técnicas de inversión. A continuación se mencionan algunas que actualmente tienen en funcionamiento este tipo de sistemas [5].

- *J4 Capital* es una compañía de expertos en trading algorítmico, utilizan la inteligencia artificial para asesorar en cuanto a estrategias de inversión [6]. Ofrecen sistemas automáticos de trading, para ayudar a los inversores a obtener resultados como los de los sistemas profesionales. Todas las operaciones las realizan cuatrocientos ordenadores, a un ritmo de un cuatrillón por segundo, que analizan datos del mercado.
- *Greenkey Technologies* utiliza procesamiento tanto de audio como texto para analizar en menor tiempo información encontrada en conversaciones o datos financieros. De esta forma, los inversores pueden acceder a notas, percepciones o tendencias del mercado en tiempo real [7].

- *Kavout* tiene un producto llamado “K Score” que procesa una gran cantidad de datos y mediante el uso de modelos predictivos realiza un ranking de diferentes acciones. Con la ayuda de inteligencia artificial, esta compañía recomienda diariamente las mejores acciones en las que invertir.
- *Equbot* analiza datos tradicionales como informes financieros, así como menos convencionales como artículos en periódicos o información de redes sociales. Utilizando Watson, el producto de IBM, las inversiones son automatizadas.

1.3. Objetivos del proyecto

En este proyecto se explora una de las diferentes áreas de inteligencia artificial aplicada a mercados financieros. En concreto, a una compraventa de activos financieros inteligente, apoyándose en técnicas de aprendizaje profundo centradas en visión artificial. El objetivo principal será entrenar un modelo propio de Deep Learning capaz de identificar correctamente oportunidades tanto de venta como de compra de activos para poder ser utilizado como una herramienta de inversión inteligente.

Este trabajo está basado en el artículo *Trading via Image Classification (JP Morgan)* [8]. Los autores de este artículo se basan en el hecho de que los agentes de inversión toman decisiones sobre la compraventa de activos mientras observan en una pantalla imágenes de gráficas de la evolución del valor de los activos. Partiendo de esta observación, formulan la hipótesis de que tratar los problemas de predicción de señales de compraventa de activos financieros como un problema de visión artificial en lugar de un problema de series temporales numéricas podría aportar valor y tener mayor éxito a la hora de recuperar dichas señales.

En su estudio, analizan los datos de todas las acciones del índice S&P 500 del periodo de 2010 a 2017. Comparan tres indicadores binarios diferentes conocidos, correspondiendo a cada uno una serie de normas algebraicas que determinan los diferentes momentos en los que habría una oportunidad compra, basándose únicamente en el valor de cierre diario de las acciones. Los tres algoritmos utilizados son: Bandas de Bollinger (BB), Convergencia Divergencia de la Media Móvil (MACD) e Índice de Fuerza Relativa (RSI).

Los datos utilizados corresponden a los valores de apertura, cierre, máximo y mínimo diarios (OHCL). En el estudio comparan los resultados de utilizar cinco tipos de representaciones diferentes de los datos, desde representar únicamente el valor del precio de cierre, hasta representar los cuatro valores OHCL incluyendo diferentes maneras de representar una noción del tiempo. De las cinco, la representación que incluye únicamente la curva del precio de cierre es la que obtiene mejores resultados, ya que aunque los otros tres datos del precio diario, así como los datos de los días anteriores, influyen en el valor de precio de cierre y por tanto podrían ser relevantes, el precio de cierre es el único valor que los algoritmos tienen en cuenta a la hora de determinar si un día presenta una oportunidad de compra o no.

Estos algoritmos tienen en cuenta los datos correspondientes a una ventana de tiempo de entre 20 y 27 días, dependiendo de la fórmula utilizada en cada uno. Por ello, las gráficas generadas incluyen la representación del día para el que se calcula el resultado del algoritmo y los correspondientes días anteriores que influyen en ese resultado.

Con las gráficas generadas, entrenan 16 modelos de aprendizaje supervisado diferentes con el objetivo de estudiar el rendimiento global de todos los modelos, no siendo el objetivo elegir el mejor modelo. La resolución de imágenes óptima resulta ser de 30x30. Entrenan todos los modelos con un total de 10 000 muestras, 5 000 correspondientes a días que presentan oportunidad de compra, y 5 000 días en los que no hay ninguna señal. Este entrenamiento se realiza tres veces, una para cada algoritmo diferente, ya que los días que presentan señales serán en muchos casos diferentes para cada uno.

Para evaluar el rendimiento de los modelos una vez entrenados, para cada día del año 2018, generan una imagen con las curvas correspondientes a los datos de los valores diarios de las acciones de dicho día y de los días anteriores relevantes, igual que para las imágenes del entrenamiento. Para cada imagen predicen si a ese día correspondería una señal de compra o no. La precisión global obtenida a la hora de detectar las señales para los algoritmos de BB y RSI es superior al 90%, y entorno al 80% en el caso del algoritmo MACD. En la Ilustración 1 se muestran los resultados obtenidos para una de las acciones en el caso del algoritmo de las Bandas de Bollinger. Se representan las señales reales así como las predichas, la mayoría de las cuales coinciden, y las que no, se encuentran en posiciones muy cerca de las condiciones que cumplirían el algoritmo.

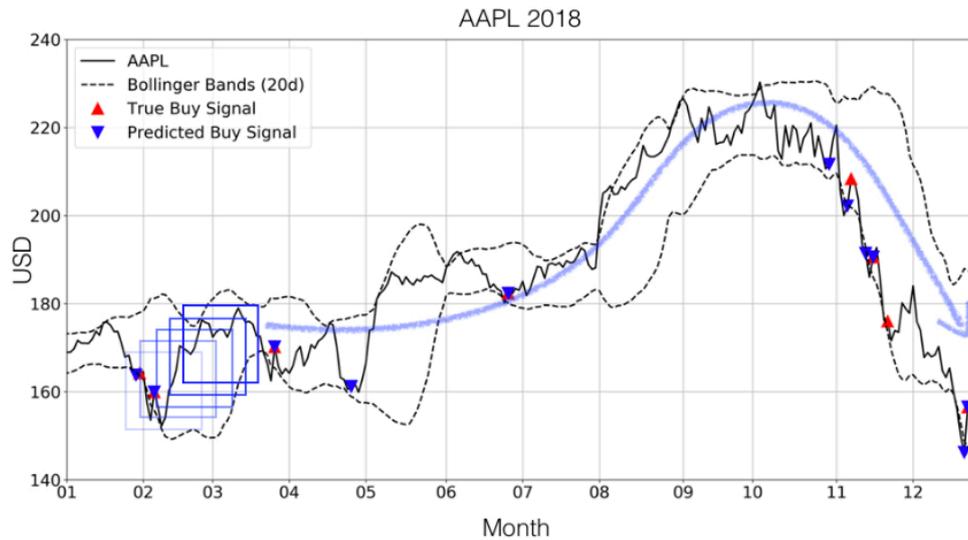


Ilustración 1. Resultados de la predicción de señales de venta de JP Morgan

Se puede concluir a partir de estos resultados, que la transformación del análisis de series temporales en un problema de visión resulta beneficioso a la hora de trabajar con datos financieros para identificar oportunidades de compra de activos. Los resultados obtenidos demuestran que tiene sentido tratar este tipo de problemas, que en el día a día son resueltos visualmente por personas, como tareas de visión artificial, imitando así la realidad.

De acuerdo con los objetivos de desarrollo sostenible, este trabajo estará alineado con el objetivo 9: industria, innovación e infraestructuras. Una de las claves de este proyecto se centra en la innovación, introduciendo una nueva forma de aplicar la tecnología y haciendo un uso eficiente de recursos. [9]

Partiendo de los conceptos y métodos descritos en este apartado, y del éxito demostrado, este trabajo se divide en cuatro objetivos principales:

1. En primer lugar, se buscará replicar los resultados del estudio. En concreto, por su utilidad para resolver problemas de visión artificial, se utilizarán modelos de redes neuronales convolucionales de entre todos los analizados en el artículo. Las imágenes que recibirá el modelo serán las correspondientes a la evolución del valor de las acciones de varias de las empresas del IBEX 35, en el periodo de tiempo de 2010 a 2017. Se evaluará la precisión con los datos de las mismas empresas para el año 2018.

El algoritmo financiero escogido para la toma de decisiones de compraventa es el de las Bandas de Bollinger. Este algoritmo de compone de tres curvas, calculadas a partir del precio de cierre de cada acción. La primera curva es una media móvil, calculada para una ventana de tiempo de 20 días. A la media móvil, se suma y resta dos desviaciones estándar (calculadas para los mismos 20 días), calculando así las dos Bandas de Bollinger, una superior y una inferior. El corte de la curva del precio con la curva superior representará una oportunidad de venta, y con la curva inferior, de compra. En este trabajo, a diferencia del estudio realizado por los expertos de JP Morgan, se trabajará tanto con las oportunidades de compra así como las de venta, siguiendo los mismos principios.

2. Una vez conseguidos unos resultados similares, se realizarán diferentes ampliaciones al estudio, añadiendo complejidad para conseguir una aplicación más práctica, así como comprobar la eficacia y versatilidad de estos modelos para distintas tareas y configuraciones de los datos. Se realizarán dos cambios diferentes en los datos de entrada para evaluar el rendimiento del modelo en casos en los que se aporte menor información. En primer lugar, de las gráficas iniciales se eliminará el último día representado, al que corresponde la acción a realizar y por la cual ha sido clasificada la imagen. De esta forma, se realizará una tarea de predicción: el modelo, conociendo los datos de los 19 días anteriores, deberá predecir la acción a realizar el día siguiente. La segunda variación consiste en eliminar de las gráficas las curvas correspondientes al algoritmo de las Bandas de Bollinger. En este caso se buscará que el modelo aprenda solo el algoritmo y sea capaz de clasificar las imágenes correctamente conociendo únicamente el valor del precio de cierre de las acciones.
3. En tercer lugar, se entrenará un modelo convolucional, que recibirá las mismas imágenes que el primer modelo, para realizar una tarea de regresión, con el objetivo de identificar el punto exacto de corte de la curva del precio con una de las Bandas de Bollinger.
4. Por último, se realizará una comparación entre los resultados obtenidos con el modelo convolucional, siendo los datos de entrada imágenes, y los resultados obtenidos con

modelos de machine learning clásicos, que toman como entrada datos numéricos en forma tabular. El objetivo de esta comparación es evaluar la utilidad y viabilidad de convertir este problema financiero de análisis temporal en uno de visión artificial, así como destacar las diferencias entre unos modelos y otros, las ventajas y desventajas de cada uno de ellos.

2. Estado del arte

2.1. Aprendizaje profundo

El aprendizaje profundo (*Deep Learning*), es una rama de *machine learning* basado en redes neuronales, cuyo aprendizaje puede ser tanto supervisado como no supervisado. Los algoritmos están inspirados en la estructura y funcionamiento del cerebro. Entre las diferentes arquitecturas del aprendizaje profundo se encuentran las redes neuronales profundas, las redes neuronales convolucionales o las redes neuronales recurrentes, y son aplicadas a diferentes campos como la visión artificial, procesamiento de audio y tareas de traducción [9].

Este tipo de modelos se consideran de aprendizaje “profundo” debido al uso de numerosas capas en la arquitectura de los modelos, que permiten alcanzar un mayor nivel de abstracción. Por este hecho, las redes profundas resultan ser más eficientes y permiten ser empleadas en aplicaciones más prácticas que los modelos clásicos, optimizando su implementación.

Las técnicas de *machine learning* clásicas presentan limitaciones a la hora de trabajar con datos que no estén procesados de antemano. El uso de estas técnicas requiere un amplio conocimiento de las mismas, siendo necesario diseñar a mano los métodos para extraer la características relevantes para la aplicación que se esté llevando a cabo. En el caso de los modelos de aprendizaje profundo, las diferentes capas que aprenden las cuáles son características que determinarán el resultado del modelo, no son diseñadas por humanos, si no que las aprende el propio modelo sin necesidad de intervención humana.

Gracias al aprendizaje profundo se han podido realizar grandes avances en diversas áreas en las hasta ahora la inteligencia artificial no había conseguido obtener resultados. Con el tiempo se obtendrán mejores resultados ya que al no requerir a penas procesamiento manual, el rendimiento de estos modelos mejorará con el aumento de la capacidad computacional y de la cantidad de datos disponibles. En *machine learning*, sin embargo, se alcanza un máximo en el rendimiento para un número de datos determinado [10].

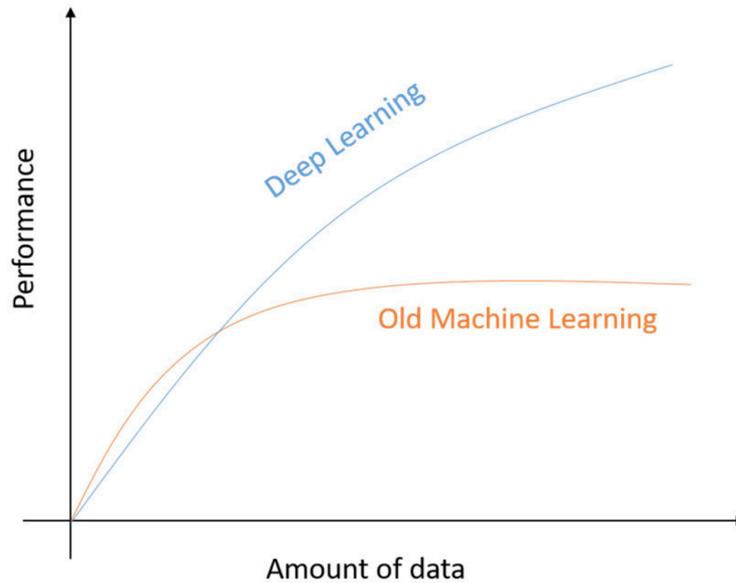


Ilustración 2. Comparación del rendimiento de modelos de deep learning frente a modelos de machine learning en función del número de datos [11]

Una de las razones por las cuales el aprendizaje profundo ha tenido éxito en los últimos años, aumentando su uso en numerosas aplicaciones, es el aumento de la capacidad computacional en comparación con la capacidad disponible hasta ahora. Este hecho, unido a una mayor facilidad para obtener grandes cantidades de datos que antes no estaban disponibles, han sido las principales contribuciones gracias a las cuales el aprendizaje profundo es tan utilizado hoy en día.

Actualmente, las principales aplicaciones son de aprendizaje supervisado, ya que estos modelos resultan ser muy eficientes en casos en los que los datos se encuentran etiquetados. Los diferentes algoritmos combinan los datos proporcionados con sus correspondientes etiquetas para realizar predicciones. Las principales tareas del aprendizaje supervisado son la regresión (predicción de un valor numérico) y la clasificación (dividir elementos en grupos predefinidos). El entrenamiento de estos modelos se realiza mediante la minimización continua del error entre la predicción y el valor real de la etiqueta.

Muchos expertos coinciden en que debido a la gran cantidad de datos sin etiquetar que hay disponibles, pronto se verán más beneficios de el aprendizaje no supervisado. Este tipo de aprendizaje se considera más complejo que el supervisado, ya que el modelo debe aprender a trabajar con los datos sin conocer el resultado esperado. Los modelos reciben los datos sin sus

etiquetas, y el resultado es una agrupación. Los modelos encuentran características similares y diferenciadoras entre los datos para así dividirlos en diferentes grupos. Este tipo de modelos ayudan a descubrir patrones desconocidos en los datos, que posteriormente podrán ser utilizados para etiquetarlos.

Otro tipo de aprendizaje profundo son los modelos de aprendizaje por refuerzo, basados en un sistema de recompensas tanto positivas como negativas, con el objetivo de alcanzar un resultado al que no se sabe como llegar.

2.2. Redes neuronales convolucionales

Una de las áreas de la inteligencia artificial es la de visión artificial. El objetivo es conseguir que los ordenadores sean capaces de ver y percibir el mundo de manera similar a como lo hacen los humanos. Los avances en visión artificial mediante aprendizaje profundo se han conseguido en gran parte gracias a las redes neuronales convolucionales.

Una red neuronal convolucional es un algoritmo de *Deep Learning* que recibe una imagen como entrada. El pre-procesamiento que requieren estas redes es mucho menor comparado con otros algoritmos de visión artificial en los que los filtros son optimizados a mano. En las redes convolucionales, con suficiente entrenamiento la red es capaz de aprenderlos sola [12]

Las redes convolucionales son capaces de capturar con éxito las dependencias espacio-temporales presentes en imágenes a través de la aplicación de filtros adecuados. La función de estas redes es reducir las imágenes para que sean más fáciles de procesar, sin perder las características importantes para conseguir buenas predicciones. Para ello hay diferentes tipos de capas, que en muchas ocasiones siguen la estructura presentada en la Ilustración 3. Los modelos suelen presentar uno o más conjuntos de capas de convolución más maxpool, seguidos de una o más capas densamente conectadas.

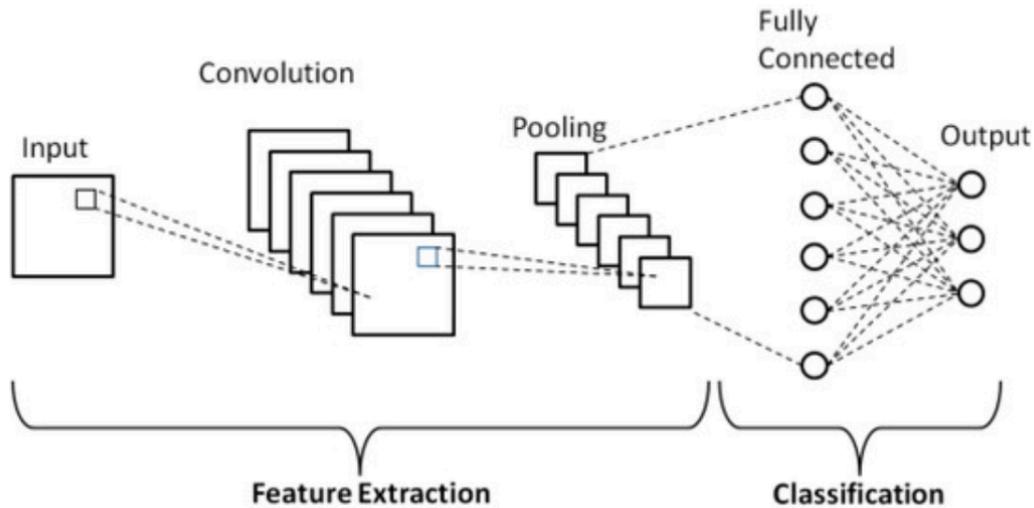


Ilustración 3. Arquitectura común de una red neuronal convolucional [13]

Las capas convolucionales son las principales en este tipo de redes. Realizan una operación de convolución, que consiste en multiplicar los píxeles de la imagen por un filtro (una matriz de menor tamaño que la imagen). Esta multiplicación se realiza elemento a elemento, y el resultado será la suma de los valores obtenidos en la multiplicación, que corresponderá a una posición en la nueva imagen. El filtro se aplica a todas las posiciones de la imagen empezando por la posición (0,0) y moviéndose hacia la derecha y hacia abajo saltando un número de posiciones a elección. Este proceso se repite hasta llegar al último píxel de la imagen. En las imágenes RGB, que al contar con tres canales tienen profundidad, los filtros tendrán la misma profundidad y el resultado de los tres canales se suma.

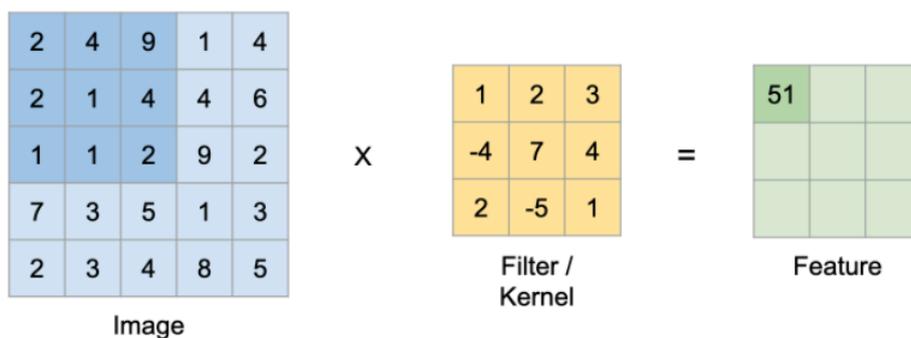


Ilustración 4. Ejemplo de operación de convolución [14]

Como resultado de esta operación, las imágenes reducen su tamaño. Para evitar este efecto se puede aplicar a las imágenes un *padding*, es decir, se añaden píxeles (habitualmente con valor igual a cero) en los bordes de las imágenes para reducir su tamaño en menor medida, mantenerlo igual después de la convolución, o incluso aumentarlo. El tamaño resultante de la imagen sigue la siguiente fórmula (asumiendo imágenes y filtros cuadrados):

$$Output = \frac{n + 2 * p - f}{s} + 1 \quad (1)$$

Donde:

- n: número de píxeles de la imagen original
- p: valor del *padding*
- f: tamaño del filtro
- s: *stride* (número de posiciones que avanza el filtro entre cada multiplicación)

El objetivo de la primera capa convolucional suele ser extraer las características de bajo nivel de la imagen, y en las capas sucesivas se extraen características cada vez más complejas a partir de las detectadas primero. Por ejemplo, en la primera capa se suelen buscar características como bordes o colores. En las siguientes capas, se detectarían formas más completas, hasta clasificar objetos enteros en las últimas capas.

Las capas convolucionales típicamente van acompañadas de una función de activación ReLU (Unidad Lineal Rectificada). Esta función evita que los resultados de la convolución puedan ser negativos, siguiendo la siguiente fórmula:

$$ReLU(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (2)$$

La representación gráfica de esta función se muestra en la Ilustración 5. La razón por la cual esta activación es tan utilizada en redes neuronales convolucionales se debe a que los modelos

suelen ser más fáciles de entrenar y habitualmente de esta manera se obtienen mejores resultados.

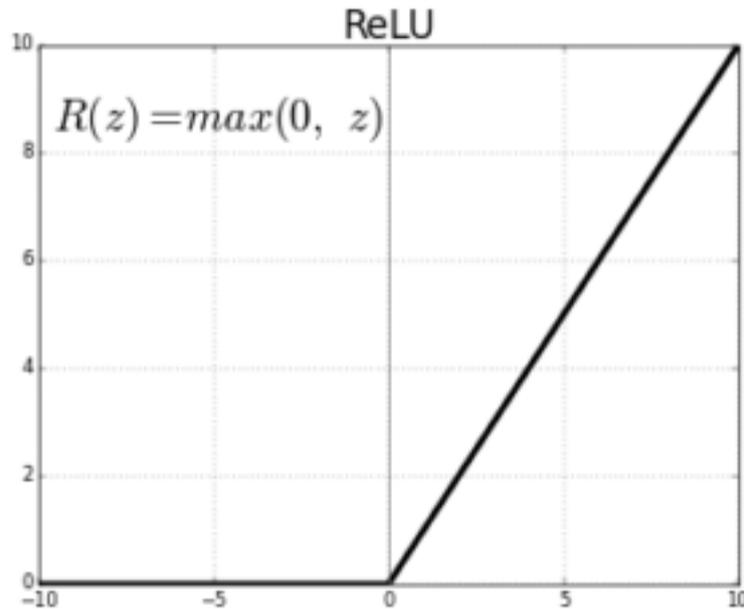


Ilustración 5. Representación de la función ReLU [15]

Las capas de *Max Pooling* tienen como función principal reducir el tamaño de la imagen resultante de la convolución, extrayendo únicamente las características más relevantes. Además, al reducir la dimensión de la imagen, disminuye el coste computacional necesario para procesarla en la siguiente capa. La operación es la siguiente: se define un tamaño para el *pooling*, que suele ser de 2×2 , y empezando por la esquina superior izquierda, se divide en matrices del tamaño del *pooling* escogido, y de cada una de esas matrices únicamente se conserva el pixel de mayor valor. En el caso de un *pooling* de tamaño 2×2 , la imagen se reduce a la mitad. El *Average Pooling* realiza la media de los pixeles de cada sub-matriz y conserva ese valor.

Después de aplicar un número adecuado de capas convolucionales y capas de Pooling, se “estira” la matriz resultante en un vector, y es común aplicar una o varias capas densamente conectadas para extraer relaciones no lineales. Por último, dependiendo del objetivo de cada modelo, se aplica una última capa que cuenta con un número de unidades correspondiente al número de salidas del modelo, en función de la aplicación la función de esta capa puede ser: lineal para una regresión, softmax para clasificación etc.

Es fácil encontrar un gran número de arquitecturas diferentes listas para ser entrenadas, desarrolladas por diferentes personas que han demostrado ser exitosas a la hora de realizar diferentes tareas de visión artificial, o en muchos casos, ganado competiciones. Estas arquitecturas pueden ser utilizadas tal cual están diseñadas, o utilizar como base para desarrollar una arquitectura adaptada a una tarea en concreto. Algunas de estas arquitecturas, de las más conocidas y comúnmente explicadas en cursos de redes convolucionales son: LeNet-5, AlexNet y VGG-16.

3. Metodología

En este apartado se detallan los métodos llevados a cabo para el desarrollo del proyecto, así como los recursos empleados.

El primer paso para poder entrenar los diferentes modelos hasta obtener los resultados buscados consiste en adquirir los datos necesarios, en este caso, los datos del precio de cierre diario de diferentes acciones del IBEX 35 para el periodo de tiempo de 2010 a 2018. Los datos serán procesados para su representación gráfica, consiguiendo así los datos de entrada que recibirán los modelos convolucionales. A partir del valor de la evolución del precio, mediante un algoritmo financiero, se clasificarán los diferentes días de dicho periodo de tiempo para cada acción según presenten una oportunidad de compra o venta, o ninguna de ellas. Esta clasificación será la salida de los modelos.

Una vez obtenidos y procesados los datos, se entrenarán diferentes modelos convolucionales según los datos de entrada, dependiendo del objetivo de cada uno, como ha sido descrito en la introducción. Para todos los modelos se partirá de una misma estructura, y se iterarán los diferentes hiperparámetros hasta obtener el mejor resultado para cada uno.

Asimismo, se entrenarán tres modelos de machine learning clásicos. Para entrenar estos modelos, en lugar de procesar los datos para conseguir imágenes, se utilizarán los datos numéricos adquiridos directamente.

Tanto el procesamiento de datos como el entrenamiento de modelos se realizarán con el lenguaje de programación Python.

3.1. Adquisición y procesamiento de datos

Como ya se ha comentado, los datos de partida con los que se trabajará corresponden al valor del precio de cierre diario de diferentes acciones del IBEX 35 del año 2010 al año 2017 para entrenamiento, y el año 2018 para evaluar. Estos datos en un primer lugar se obtendrán en

formato numérico, adquiriendo así el valor correspondiente para cada acción para cada día del periodo de tiempo indicado.

Para obtener estos datos, se empleará el paquete de Python `investpy` [16]. Este paquete permite extraer los datos históricos de la página web *Investing.com*. Los datos pueden ser de acciones, fondos, monedas o índices entre otros. Los datos obtenidos pueden ser guardados en variables de Python con las diferentes funciones con las que cuenta el paquete. Para este proyecto se busca obtener el valor del precio de cierre de diferentes acciones durante un periodo determinado. Con la función `get_stock_historical_data` que incluye el paquete, basta con indicar el nombre de la acción, el país para el cual se quieren obtener los datos y la fecha de comienzo y fin. A continuación se muestra un ejemplo:

```
data = investpy.get_stock_historical_data(stock='BBVA',
                                         country='spain',
                                         from_date='01/01/2010',
                                         to_date='31/12/2017')
```

En este caso se obtendrían los datos correspondientes a la acción BBVA en España, en el periodo de 2010 a 2017. Para cada día en que haya datos disponibles se guardarían en la variable *data* los valores del precio de apertura, el máximo, el mínimo y el cierre, así como el volumen y la moneda correspondientes.

En el caso de este trabajo solo se guardarán los datos correspondientes al precio de cierre, ya que es la evolución de estos valores lo único que se tendrá en cuenta a la hora de identificar oportunidades de compraventa.

Las acciones del IBEX 35 para las cuales se encuentran disponibles los datos en *Investing*, y por tanto las que se tendrán en cuenta para obtener los datos necesarios se muestran en la Tabla 1:

UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
MÁSTER EN INGENIERÍA INDUSTRIAL

1. ACS	7. Inmobiliaria Colonial	13. Naturgy Energy
2. Acerinox SA	8. Ferrovial	14. Red Eléctrica
3. Acciona	9. Iberdrola	15. Repsol
4. Banco de Bilbao Vizcaya	10. Indra Systems	16. Banco Santander
5. Bankinter	11. Inditex	17. Telefónica
6. Caixabank	12. Mapfre	18. Mediaset España Comunicación

Tabla 1. Acciones del IBEX 35 disponibles en el paquete Investpy

Una vez obtenidos los datos, el siguiente paso consistirá en identificar las oportunidades de compra y de venta en ese periodo de tiempo. Para ello se utilizará el conocido algoritmo de las Bandas de Bollinger. Esta herramienta fue diseñada por John Bollinger, y tiene como objetivo identificar cuando un activo está sobrecomprado o sobrevendido [17].

Las bandas se componen de tres curvas: una media móvil simple, y dos bandas, una superior y una inferior, que se sitúan a dos desviaciones estándar (negativa y positiva) del la media móvil. La media móvil suele ser calculada para un periodo de 20 días.

Las diferentes curvas se calculan de la siguiente manera. Para la media móvil, se calcula la media del precio de cierre de los primeros 20 días para obtener el primer punto. Para los puntos consecutivos se mueve la ventana de tiempo un día, incluyendo el día siguiente al periodo anterior y dejando de tener en cuenta el primero. Para cada punto de la curva de la media, se suman y restan dos desviaciones estándar, calculada para el mismo periodo de 20 días, los valores resultantes son los correspondientes a las Bandas de Bollinger. La desviación estándar mide el nivel de varianza de la media.

Las fórmulas para obtener las curvas se muestran a continuación:

$$\text{Banda Superior} = \text{MediaMovil}(\text{Precio}, 20 \text{ días}) + 2 * \sigma(\text{Precio}, 20 \text{ días}) \quad (3)$$

$$\text{Banda Inferior} = \text{MediaMovil}(\text{Precio}, 20 \text{ días}) - 2 * \sigma(\text{Precio}, 20 \text{ días}) \quad (4)$$

La representación de las curvas para el periodo de 2018 para la acción BBVA es la siguiente:

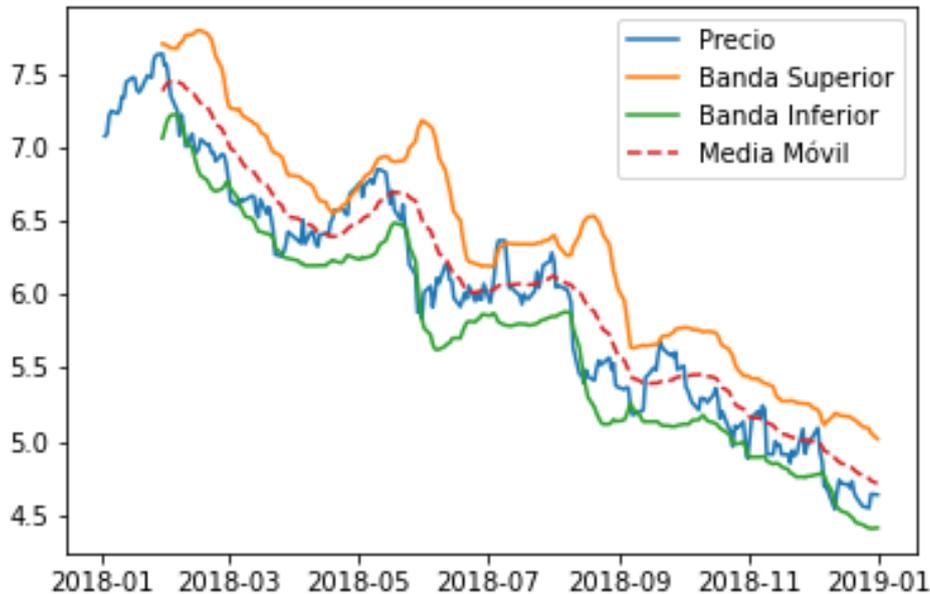


Ilustración 6. Ejemplo de representación de las Bandas de Bollinger para la acción BBVA durante el año 2018

A partir de la representación gráfica de las bandas se pueden sacar varias conclusiones. Cuanto más cerca se encuentre el precio de la banda superior, más sobrecomprada estará la acción, y al revés para la banda inferior. Las bandas también pueden ser utilizadas para medir la volatilidad de la acción: cuanto mayor sea la distancia entre las bandas, mayor será la volatilidad del activo, ya que esto depende de la varianza (desviación estándar), que en este caso será elevada.

Para identificar las diferentes oportunidades de compra o venta, este trabajo se basará en el concepto de reversión de la media [18]. Esta estrategia se apoya en la idea de que el precio siempre volverá a la media, por ejemplo, si el precio ha subido mucho, lo más probable es que vuelva al nivel medio. En el caso de las Bandas de Bollinger, el 90% del tiempo el precio se encuentra entre las dos bandas, por ello, el cruce del precio por encima de la banda superior o debajo de la banda inferior se considera un hecho importante. De acuerdo con la teoría de reversión a la media, estos cruces deberían ser seguidos por una vuelta a valores ‘normales’.

Basándose en estos conceptos, las señales de compra y venta en este trabajo se calcularán de la siguiente manera: cuando la curva del precio corte la banda inferior por arriba, se generará

una señal de compra, ya que cuando el precio se encuentre por debajo de la banda inferior, se puede considerar que el activo está sobrevendido, en el momento en que el precio vuelve a cruzar con la banda inferior, volverá a acercarse a la media. Las señales de venta se generan cuando el precio corta con la banda superior por debajo, siguiendo los mismos principios que las señales de compra, cuando el mercado esta sobrecomprado.

Se ha optado por emplear esta estrategia de inversión siguiendo el estudio realizado por JP Morgan que tarta de replicar este trabajo, pero no se recomienda usar este sistema de manera independiente, si no como información orientativa. John Bollinger recomienda combinarlo con otros indicadores no correlados para obtener información más completa a cerca del mercado.

Estas señales son las que determinarán los datos que serán utilizados para entrenar los diferentes modelos. En primer lugar, para cada acción y durante todo el periodo de tiempo para el que se han extraído los datos, se calcularán las curvas de la desviación estándar y ambas bandas con las fórmulas indicadas anteriormente. A continuación, se obtendrán todas las señales de compra identificadas, calculando los momentos en los que el precio cruza por arriba de la banda inferior, y se guardarán estos índices en un vector. Lo mismo para las señales de venta, cuando el precio cruza por debajo de banda superior. Para cada uno de estos índices, se generará una gráfica que posteriormente será guardada como formato de imagen .jpg. Cada gráfica tendrá una ventana de tiempo de 20 días, siendo el último día el correspondiente a la señal generada, representando también las curvas de los 19 días anteriores, ya que serán esos 19 días los que influyen en el valor de las bandas en el día correspondiente a la señal. En las imágenes se incluirá la curva del precio, la media, y las bandas. La Ilustración 7 y la Ilustración 8, muestran un ejemplo de imagen correspondiente a un día en el que habría una señal de compra, y uno con señal de venta, respectivamente.

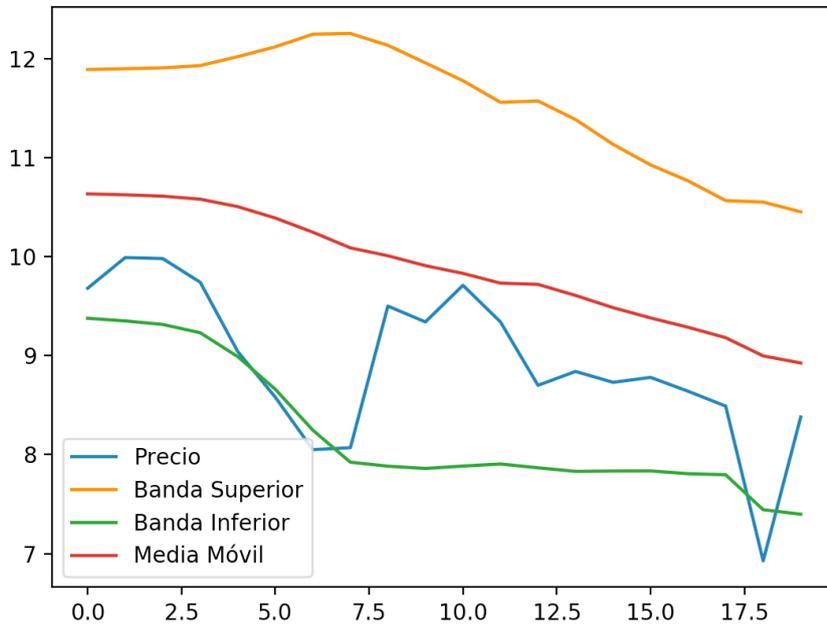


Ilustración 7. Muestra de imagen correspondiente a una señal de compra

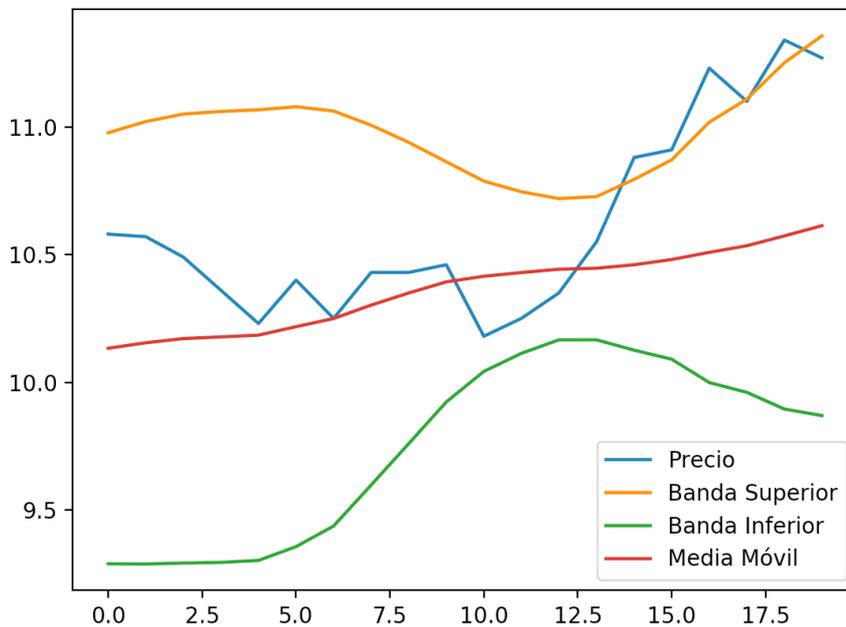


Ilustración 8. Muestra de imagen correspondiente a una señal de venta

Las imágenes guardadas que serán utilizadas para entrenar los modelos no incluyen leyenda, y serán recortadas para no incluir el valor de los ejes. La resolución de las imágenes será de

50x50 DPI antes de ser recortadas, y finalmente, las imágenes que serán utilizadas tendrán una resolución de 37x37 píxeles tras recortar los ejes.

Una vez calculadas todas las señales de compra y de venta para todas las acciones y las han sido imágenes guardadas, se generarán las imágenes correspondientes a los días en los que no se identifica ninguna señal. Para obtener un set de datos de entrenamiento lo más balanceado posible, se generará la mitad de imágenes del total de las correspondientes a ambas señales, para que los datos contengan aproximadamente un tercio de cada tipo de señal (o no señal). Los días para los que se generarán imágenes de no señal se elegirán al azar hasta llegar al número de muestras escogido. En la Ilustración 9, se muestra un ejemplo de imagen clasificada como ‘sin señal’.

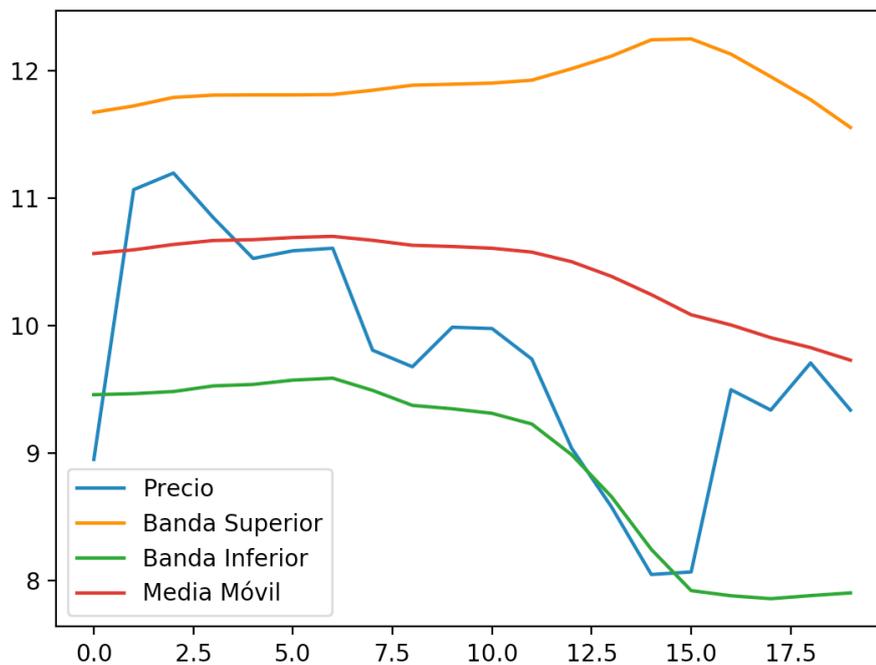


Ilustración 9. Muestra de imagen correspondiente a un día que no presenta señal

En total se trabajará con 4212 muestras, 1354 serán señales de compra, 1196 señales de venta, y 1272 sin señal. Las imágenes servirán como datos de entrada para los modelos convolucionales, cuyo objetivo será clasificarlas según el tipo de señal que representan. Las imágenes serán clasificadas con un vector binario de tres posiciones, en el que la posición en que haya un 1 indicará el tipo de imagen. En la posición 0 será muestra sin señal, en la posición 1 señal de compra, y en la posición 2 señal de venta.

El objetivo principal de los modelos será realizar una tarea de clasificación. Además se añadirá una tarea de regresión; se entrenará los modelos para identificar el punto exacto del corte de la curva del precio con una de las bandas en las muestras en las que se haya identificado una señal de compra o venta. Para esta tarea de regresión, los datos de entrada serán las mismas imágenes que para los modelos de clasificación, cambiando el vector de salida. En este caso será un vector de dos posiciones indicando las coordenadas X e Y del punto de corte, normalizándolas previamente, como se explicará más adelante en detalle.

En el caso de los modelos numéricos de machine learning, los datos de entrada no serán imágenes, si no un vector de 20 posiciones con el valor del precio durante los 20 días correspondientes a la imagen que reciben los modelos convolucionales.

3.2. Modelos neuronales convolucionales

En este apartado se analizarán diferentes modelos convolucionales, entrenados por separado para conseguir tres objetivos diferentes. Por un lado, replicando el trabajo realizado por los autores del artículo *Trading via Image Classification* detallado anteriormente, se entrenará un modelo para clasificar imágenes de 20 días según la señal correspondiente al último día representado. Las imágenes en este caso tendrán las mismas características que las representadas anteriormente, incluyendo todas las curvas.

Para estos mismos datos se entrenará un modelo para realizar una tarea regresión con el objetivo de identificar el punto de corte de las curvas, en el caso de que lo haya. Se entrenarán dos modelos de regresión diferentes, uno para cada tipo de señal.

Estos dos primeros tipos de modelo (clasificación y regresión) estarán relacionados de la siguiente manera: cada uno será entrenado por separado hasta conseguir buenos resultados, buscando maximizar la precisión en la clasificación y minimizar el valor de la función de coste en la regresión. Una vez entrenados los modelos, se realizará primero la clasificación de los datos del set de test, y según la clasificación, se empleará un modelo u otro de regresión para predecir las coordenadas del punto de corte. Es decir, si una imagen ha sido clasificada por el primer modelo como una señal de venta (pudiendo ser correcta o no la clasificación), la

predicción del punto de corte de esta imagen se hará con el modelo de regresión entrenado con los datos correspondientes a imágenes de venta. Igual para las imágenes clasificadas como compra. En el caso de las imágenes clasificadas sin señal, el valor del punto de corte se establecerá directamente en cero, sin necesidad de haber entrenado un modelo para esta clase de imágenes.

En tercer lugar, se empleará un modelo diferente para realizar una tarea de predicción. En este caso, el modelo no conocerá el valor de las curvas del día que determina la clasificación de la imagen. Para este modelo, se eliminará de las imágenes utilizadas para los modelos anteriores la representación del último día, resultando en imágenes con una ventana de tiempo de 19 días.

Para ambos modelos de clasificación, se modificarán los datos para analizar los resultados de entrenar modelos con muestras que incluyan únicamente el valor del precio de la acción. Se eliminará de las imágenes la representación de las curvas de las Bandas de Bollinger así como de la media móvil.

El proceso de entrenamiento de los modelos será el mismo para todos ellos: se partirá de la estructura indicada en el artículo de JP Morgan, con alguna modificación para ajustarse a los requisitos de este proyecto, y se realizarán iteraciones tanto en los hiperparámetros (número de particiones en la validación cruzada, número de épocas...) como en la arquitectura de la red convolucional. La estructura base de las capas de la red será la siguiente:

- Zero Padding 3x3 en la primera capa, para
- Tres capas convolucionales de 32 filtros 3x3 con activación ReLU
- Max Pooling 2x2 después de cada capa convolucional
- Una última capa con activación Softmax

En cuanto al resto de hiperparámetros:

- Optimizador: Adam
- Función de coste: entropía cruzada categórica
- Tamaño de batch: 16 muestras

- 50 épocas

Para los modelos de clasificación, la activación de la última capa será una función softmax en lugar de una sigmoide como aparece indicado en el artículo en el que se basa esta estructura. Esto se debe a que en el estudio realizado por JP Morgan se distingue entre dos situaciones: oportunidad de compra o ausencia de oportunidad, siendo la clasificación binaria. En este trabajo se ha ampliado la clasificación a una tercera posibilidad, las señales de venta. La función sigmoide suele ser utilizada en problemas de clasificación en los que se distingue únicamente entre dos clases, mientras que la función softmax es utilizada para clasificación múltiple. La función softmax no es más que una extensión de la función sigmoide, ya que si se aplica su fórmula para dos clases solo, el procedimiento es el mismo.

El resultado de una función de activación softmax es un vector de probabilidades que suman uno, este vector representa la distribución de la probabilidad de las diferentes opciones o clases. Es por esto que es ampliamente utilizada en Deep learning para tareas de clasificación. La ecuación de esta función de activación es la siguiente, siendo el vector Y el resultante de la capa anterior:

$$\text{Softmax}(Y_i) = \frac{e^{Y_i}}{\sum_j e^{Y_j}} \quad (5)$$

De este modo, la posición con la mayor probabilidad, será seleccionada como la clase predicha por el modelo.

Otro cambio respecto a la estructura original es la función de coste como entropía cruzada categórica en lugar de entropía cruzada binaria, por la misma razón por la cual se ha cambiado la función de activación de la clasificación, al ser un problema de varias clases en lugar de binario. En este caso, la clase no será representada por un único dígito, distinguiendo entre dos opciones según el valor sea cero o uno. El resultado de la predicción será un vector de tres posiciones, una por cada clase, por lo que la función de coste de entropía cruzada binaria no se ajusta al problema.

La ecuación 4 corresponde a la función de coste de la entropía cruzada categórica. El coste se calcula de la siguiente manera: se compara la distribución de probabilidad de la predicción (resultado de la capa softmax) con la distribución real. La distribución real será un vector en el que todas las posiciones son cero, excepto la correspondiente a la clase, que será uno. Cuanto más cerca se encuentre la distribución de probabilidad predicha de la real, menor será la función de coste.

$$L(y, \hat{y}) = - \sum_{j=0}^M \sum_{i=0}^N (y_{ij} * \log(\hat{y}_{ij})) \quad (6)$$

Todos los modelos serán entrenados con la librería Keras. Keras es una API para redes neuronales de código abierto escrita en Python. Se puede ejecutar sobre diferentes marcos de Deep learning, en este trabajo se utilizará TensorFlow. Esta API ofrece una manera intuitiva de entrenar redes neuronales de alto nivel, resultando más sencillo el desarrollo de modelos de aprendizaje profundo.

A continuación se muestra el código correspondiente a la estructura base detallada en Keras, así como los paquetes necesarios que se deben importar:

```
#import packages
from tensorflow import keras
from keras import layers
from keras.layers import Input, Dense, Activation, ZeroPadding2D,
BatchNormalization, Flatten, Conv2D
from keras.layers import MaxPooling2D
from keras.models import Model
from keras.preprocessing import image
from keras.utils import layer_utils
from keras.utils.data_utils import get_file
from keras.applications.imagenet_utils import preprocess_input
import pydot
from IPython.display import SVG
from keras.utils.vis_utils import model_to_dot
```

```
from keras.utils import plot_model
import keras.backend as K
import tensorflow as tf
from sklearn.model_selection import train_test_split, StratifiedKFold
from sklearn.model_selection import KFold
K.set_image_data_format('channels_last')

X_input = Input(X_train.shape[1:])

#Zero-Padding: pads the border of X_input with zeroes
X = ZeroPadding2D((3, 3))(X_input)

# CONV1 -> BN -> RELU Block applied to X
X = Conv2D(32, (3, 3), strides = (1, 1))(X)
X = BatchNormalization(axis = 3)(X)
X = Activation('relu')(X)
# MAXPOOL1
X = MaxPooling2D((2, 2), name='max_pool0')(X)

# CONV2 -> BN -> RELU Block applied to X
X = Conv2D(32, (3, 3), strides = (1, 1), name = 'conv1')(X)
X = BatchNormalization(axis = 3, name = 'bn1')(X)
X = Activation('relu', name = 'relu1')(X)
# MAXPOOL2
X = MaxPooling2D((2, 2), name='max_pool1')(X)

# CONV3 -> BN -> RELU Block applied to X
X = Conv2D(32, (3, 3), strides = (1, 1), name = 'conv2')(X)
X = BatchNormalization(axis = 3, name = 'bn2')(X)
X = Activation('relu', name = 'relu2')(X)
# MAXPOOL3
X = MaxPooling2D((2, 2), name='max_pool2')(X)
```

```
# FLatten
X = Flatten() (X)

#Classification output
X_class = Dense(3, activation = 'softmax', name='class_output') (X)

# Create model
model = Model(inputs = X_input, outputs = X_class, name='Model')

# Compile model
model.compile(optimizer = 'Adam', loss = "categorical_crossentropy" , metrics =
["categorical_accuracy"])

# Fit model to the training data
model.fit(X_train, Y_train, epochs = 50, batch_size = 16)

# Evaluate model with test data
model.evaluate(X_test, Y_test)
```

3.2.1. Modelo de clasificación de señales de compraventa pasadas

El primero modelo entrenado tiene como objetivo principal evaluar la efectividad de una red neuronal convolucional a la hora de recuperar las señales de compra y venta cuando los datos de entrada son imágenes. Este modelo busca clasificar correctamente imágenes que contienen datos históricos pasados. Este modelo no tiene como finalidad su uso en un entorno real, ya que se apoya en conocer de antemano el valor del precio de cierre de la acción del día para el cual se quiere identificar una oportunidad.

Como se ha detallado anteriormente, los datos que recibe este modelo consisten en un vector de 4212 muestras, cada muestra siendo una imagen de tamaño 37x37x3. Se entrenarán dos tipos de modelo diferentes: en uno de ellos, los datos que recibe el modelo incluyen la representación de todas las curvas que influyen en la generación de señales: precio de cierre, media móvil y las bandas inferior y superior de Bollinger. El segundo modelo solo incluirá el precio de la acción.

La Tabla 2 muestra los resultados de diferentes modelos entrenados para las imágenes que contienen la representación de las cuatro curvas, iterando diferentes hiperparámetros buscando conseguir la máxima precisión posible.

MODELO	FOLDS	EPOCHS	ACCURACY				
			Total	Media	No Signal	Buy	Sell
1	5	50	0,522	0,821	0,497	0,993	0,974
2	10	50	0,761	0,893	0,751	0,927	1
3	10	100	0,845	0,800	0,851	0,628	0,922
4	10	75	0,685	0,874	0,67	0,964	0,987
5	10	150	0,782	0,889	0,773	0,92	0,974
6	10	125	0,752	0,897	0,741	0,964	0,987

Tabla 2. Resultados de precisión para los modelos convolucionales de clasificación de señales pasadas, para imágenes con la representación de todas las curvas

Los dos hiperparámetros principales que se han iterado buscando mejorar el rendimiento del modelo son el número de particiones (*folds*) de la validación cruzada y el número de épocas (*epochs*). Otros hiperparámetros, como el tamaño de los filtros, al ser modificados respecto a la estructura base detallada anteriormente, presentaban resultados considerablemente peores, por lo que no se han incluido en la comparación.

Para cada modelo, se presentan cinco medidas de la precisión diferentes. La primera medida, ‘total’, es el valor de la precisión medida por el modelo, para cada muestra en el set de datos de evaluación, se convierte la predicción del modelo en un vector binario, tomando como predicción la posición del vector que presenta una mayor probabilidad. La medida de precisión se calcula como el porcentaje de muestras que han sido predichas correctamente. Las últimas tres medidas se han calculado para cada tipo de muestra por separado, por ejemplo, para las muestras que corresponden a días en los que no hay ninguna señal, se ha calculado el porcentaje de las muestras que se han predicho como no señal. Lo mismo para las señales de compra y venta. Separar la medida de la precisión en estas tres categorías resulta útil para conocer más

información a cerca de como actúa cada modelo, y poder comparar la precisión para un tipo de muestras y otras, y observar si existe una diferencia considerable entre ellas. Por último, a partir de las medidas de precisión para cada clase, se calcula la media, que no es equivalente al total ya que la proporción de cada tipo de muestra no está igualada, al ser la mayor parte de las muestras del tipo sin señal. Comparando esta última medida con el valor de la precisión total se puede intuir si el valor total de la precisión sirve como buena medida para conocer la precisión de todas las muestras (si ambas medidas son similares), o si está principalmente influenciada por un tipo de muestra concreto (si hay una gran diferencia entre ambas).

En todos los modelos, excepto para el modelo 3, la precisión conseguida para las muestras que representan algún tipo de señal es bastante más alta que para las muestras que no representan señal. Al ser la mayor parte de las muestras de set de evaluación del tipo no-señal, la precisión total se ve mayormente afectada por la precisión en la clasificación de este tipo de muestras. Hay muchas más señales de este tipo ya que como se ha comentado anteriormente, al representar las Bandas de Bollinger, el 90% de la actividad tiene lugar entre las bandas, por lo tanto, los días en los que se identifican oportunidades de compra o venta, son la minoría. Es por esta razón que tiene sentido estudiar la precisión de cada tipo de muestra por separado.

El modelo 3, entrenado con 10 particiones y 100 épocas, presenta la precisión total más alta, de un 84,5%. A su vez, precisión media de todas las clases es del 80%, por lo que se puede concluir que la precisión total es una representación adecuada del rendimiento del modelo para la mayor parte de las muestras, siendo el modelo que presenta una menor diferencia entre la precisión total y la precisión media.

De todas las clases, la que mejores resultados obtiene en todos los modelos, con una precisión mayor al 92% en todos los casos, es la correspondiente a oportunidades de venta.

Una vez entrenados varios modelos con diferentes hiperparámetros hasta obtener el mejor resultado, se entrenará un nuevo modelo partiendo de la estructura base detallada anteriormente. La diferencia entre este modelo y el anterior se encuentra en la representación de los datos. Con este modelo se pretende añadir complejidad a la clasificación, al no representar en las imágenes de entrada al modelo las curvas pertenecientes a las Bandas de Bollinger. Este modelo busca añadir un punto de complejidad a la aplicación; en el modelo anterior, la clasificación se realizaba en función de que hubiese o no un cruce entre dos de las

curvas representadas el último día, por lo tanto, el modelo debía identificar si existía dicho punto de cruce. Este nuevo enfoque busca que la red neuronal sea capaz de intuir el algoritmo conociendo únicamente el precio. El modelo deberá aprender diferentes características de los datos, ya que en este caso no habrá un punto de corte que determine la clasificación de las imágenes.

La Ilustración 10 e Ilustración 11 muestran la diferencia entre las imágenes que reciben un modelo y otro. En este caso, la muestra corresponde a una imagen clasificada como oportunidad de compra. Para la segunda imagen, el modelo deberá ser capaz de reconocer la oportunidad de compra basándose únicamente en la evolución del precio de cierre durante los 20 días representados.

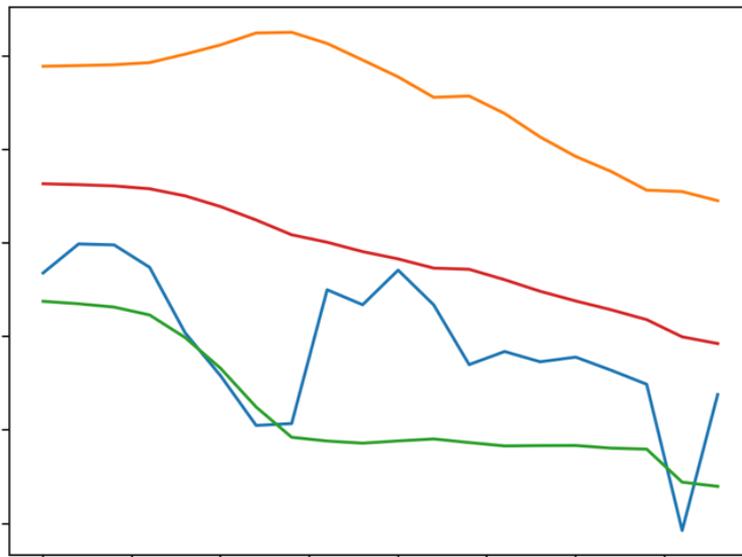


Ilustración 10. Ejemplo de muestra incluyendo la representación de todas las curvas

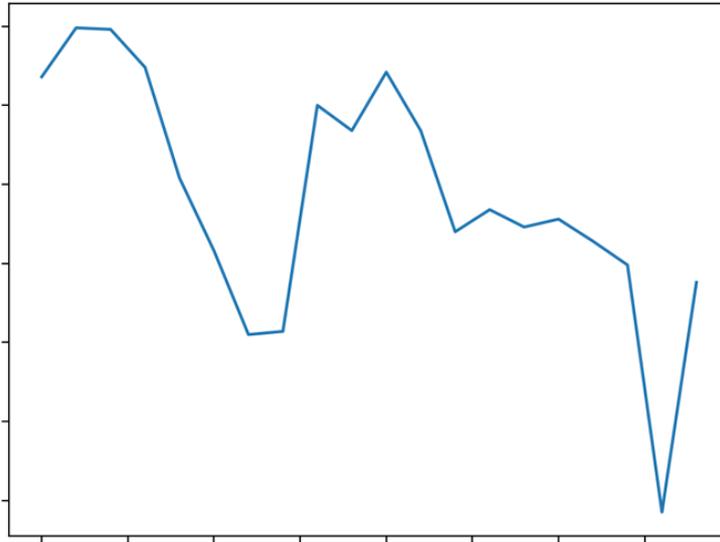


Ilustración 11. Ejemplo de muestra incluyendo la representación únicamente de la curva del precio de cierre

En la Tabla 3 se muestran los resultados del entrenamiento de diferentes modelos para este nuevo formato de los datos. Se puede observar como para estos nuevos modelos, la precisión alcanzada es superior a la conseguida en los modelos que han sido entrenado con las imágenes en las que se representaban todas las curvas. En este caso, igual que en la tabla anterior, el número óptimo de particiones es mayor que cinco, obteniendo los mejores resultados con diez particiones.

MODELO	FOLDS	EPOCHS	ACCURACY				
			Total	Media	No Signal	Buy	Sell
1	10	50	0,853	0,899	0,85	0,861	0,987
2	10	100	0,8133	0,913	0,806	0,934	1
3	5	50	0,706	0,845	0,535	1	1

Tabla 3. Resultados de precisión para los modelos convolucionales de clasificación de señales pasadas, para imágenes con la representación únicamente de la curva del precio de cierre

La precisión total más alta alcanzada es del 85,3%, en el caso del modelo 1, entrenado con 10 particiones y 50 épocas. Igual que en los modelos anteriores, la precisión es más alta en todos los modelos para las muestras que presentan uno de los dos tipos de señal que para las que no.

En la Tabla 4 se realiza una comparación de la clasificación respecto a la clase real de las muestras.

En cuanto a la clasificación de las imágenes clasificadas con señal, para ambas señales, los errores se cometen al clasificar las muestras como no-señal, sin confundir en ningún caso la clasificación una señal con otra.

		PREDICCIÓN	NO SEÑAL	COMPRA	VENTA
SEÑAL REAL	NO SEÑAL		3397	193	408
	COMPRA		19	118	0
	VENTA		1	0	76

Tabla 4. Comparación de los resultados de la clasificación frente al valor real de las muestras

3.2.2. Modelo de predicción de señales de compraventa futuras

En el apartado anterior se ha demostrado que las redes neuronales convolucionales son capaces de obtener buenos resultados a la hora de interpretar gráficas de la representación de series temporales financieras para identificar oportunidades de compra o venta en determinados días. Estos primeros modelos no tienen una aplicación práctica real, pero sirven como primer paso para evaluar la viabilidad del cambio de planteamiento de un problema numérico a uno visual.

El siguiente paso será ampliar el alcance del trabajo para conseguir desarrollar una herramienta que pueda ser útil y servir como apoyo para una estrategia de inversión efectiva. Para conseguir esto, se planteará desarrollar un nuevo modelo que sea capaz de predecir la presencia de una

oportunidad de compra o venta el día siguiente al último representado. En este caso, conocido el precio de cierre de un día concreto y de los 18 anteriores, se podrá predecir que acción sería la más óptima tomar el día siguiente. De este modo, se podrían tomar decisiones con antelación, anterior consiste en el formato de los datos de entrada. Mientras que los datos de salida (la clasificación de cada muestra) permanece igual, en este nuevo modelo las imágenes que componen los datos de entrada al modelo sólo incluirán la representación de los 19 días anteriores al día para el cual se quiere conocer la acción a realizar. Los cambios en los imágenes son dos: la ventana de tiempo se reduce un día, y la información del día que determina la clasificación de la imagen no se conoce. En este caso, al contrario que en modelo anterior no hay un punto de cruce entre la curva del precio con una de las dos bandas, por lo que la clasificación deberá realizarse a partir de la información a cerca de la evolución de las curvas, en lugar de buscar si dicho cruce está representado o no.

Un ejemplo del cambio de una muestra de un modelo a otro se puede observar en la Ilustración 12 e Ilustración 13, siendo la última la correspondiente al nuevo formato que servirá para la predicción.

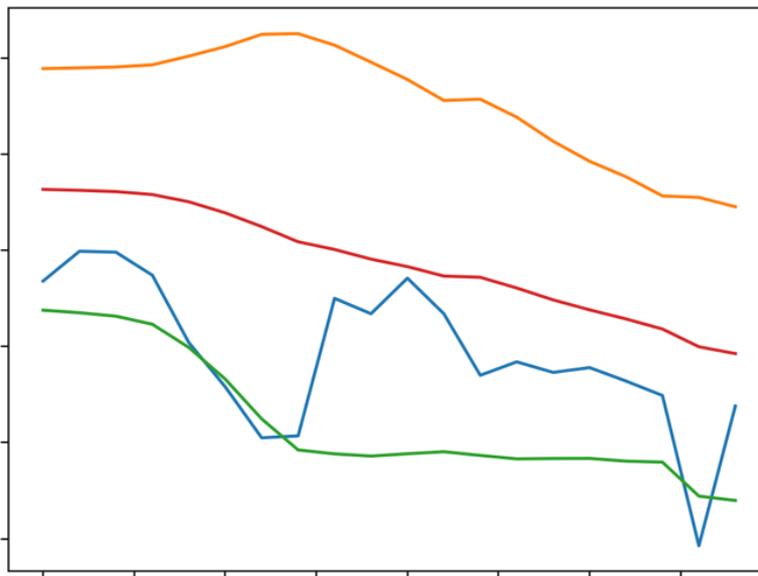


Ilustración 12. Ejemplo de muestra que incluye la representación de 20 días para todas las curvas

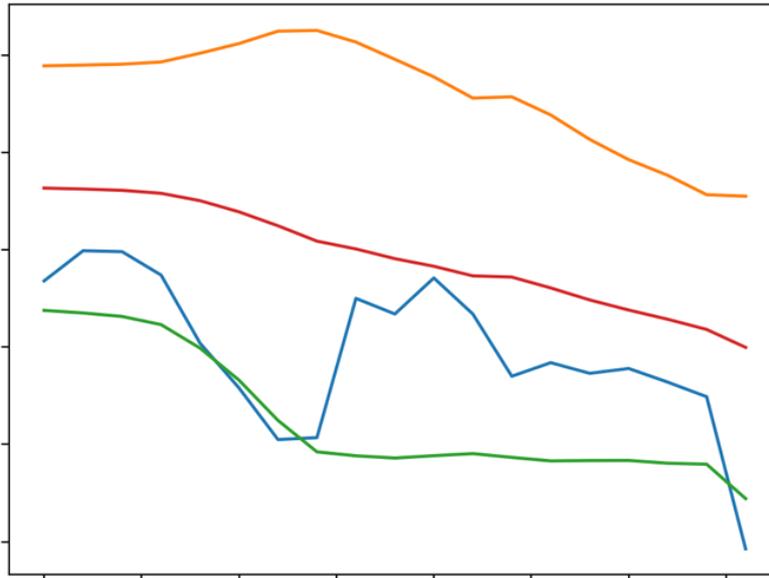


Ilustración 13. Ejemplo de muestra que incluye la representación de 19 días para todas las curvas

En la Tabla 5 se recogen los resultados de los diferentes modelos entrenados con estos datos.

MODELO	FOLDS	EPOCHS	ACCURACY				
			Total	Media	No Signal	Buy	Sell
1	5	50	0,433	0,700	0,409	0,978	0,714
2	10	50	0,504	0,789	0,481	0,898	0,987
3	10	100	0,424	0,759	0,395	0,985	0,896
4	10	150	0,535	0,774	0,516	0,883	0,922

Tabla 5. Resultados de precisión para los modelos convolucionales de clasificación de señales futuras, para imágenes con la representación de todas las curvas

El mejor modelo de este tipo a penas alcanza una precisión del 53,5%, significativamente inferior respecto al modelo equivalente para las imágenes de 20 días (84,5%). El modelo entrenado parece no ser lo suficientemente complejo para ser capaz de recuperar las señales de compra y venta correctamente, al clasificar erróneamente aproximadamente la mitad de

muestras que no presentan ninguna oportunidad como señal de venta o de compra, cerca del 25% de cada tipo, como indica la Tabla 6.

		PREDICCIÓN	NO		
			SEÑAL	COMPRA	VENTA
SEÑAL REAL	NO SEÑAL		2063	959	986
	COMPRA		12	121	4
	VENTA		5	1	71

Tabla 6. Comparación de los resultados de la predicción frente al valor real de las muestras para las imágenes que incluyen la representación de todas las curvas

Igual que en el apartado anterior, se entrenará también un modelo con imágenes que contienen sólo la curva del precio. En este caso, las imágenes no incluirán la representación de las Bandas de Bollinger ni la información a cerca del día 20, que es el día para el cuál se desea conocer la decisión óptima que tomar. De los diferentes modelos desarrollados a lo largo del trabajo, este será el que cuente con menor información para realizar la tarea de clasificación. Aumentando de esta forma la complejidad del problema no sólo convierte este modelo en una herramienta útil que pueda ser empleada a la hora de invertir, si no que también sirve para analizar la versatilidad y escalabilidad de este proyecto y de utilizar redes neuronales convolucionales para estrategias de inversión.

Los resultados de la precisión de los diferentes modelos de predicción entrenados se muestran en la Tabla 7. Estos modelos alcanzan una precisión mucho mayor a los modelos en el caso en el que se representan todas las curvas, con resultados muy similares a los modelos en los que el día 20 se incluía representado en las imágenes de entrada.

MODELO	FOLDS	EPOCHS	ACCURACY				
			Total	Media	No Signal	Buy	Sell
1	10	50	0,855	0,874	0,854	0,86	0,987
2	5	50	0,62	0,801	0,602	0,919	0,987
3	10	100	0,837	0,882	0,834	0,838	0,987

Tabla 7. Resultados de precisión para los modelos convolucionales de clasificación de señales futuras, para imágenes con la representación únicamente del precio

El modelo con el que mejor precisión se obtiene es el modelo 1, con diez particiones y 50 épocas. La precisión total es del 85,5%, y la media, 87,4%. En la Tabla 8 se recogen los resultados de la predicción de la clasificación de las muestras respecto a su valor real.

		PREDICCIÓN	NO		
			SEÑAL	COMPRA	VENTA
SEÑAL REAL	NO SEÑAL		3414	435	150
	COMPRA		19	117	0
	VENTA		7	0	70

Tabla 8. Comparación de los resultados de la predicción frente al valor real de las muestras para las imágenes que incluyen la representación únicamente del precio de cierre

3.2.3. Modelo de regresión

El tercer tipo de modelo neuronal convolucional desarrollado tiene un objetivo diferente a los anteriores, cuya finalidad consistía en clasificar correctamente las imágenes. Este modelo realizará una tarea de regresión, posterior a la clasificación.

Con este modelo, se pretende identificar el punto exacto de corte, si es que lo hay, entre la curva del precio con una de las bandas.

En primer lugar, se calcularán las coordenadas del punto de intersección para las muestras que presenten una señal de compra o de venta. La coordenada en el eje X del punto de corte siempre tendrá un valor entre 18 y 19, ya que la intersección que determina la clase de cada muestra siempre se da en el último día (el último día corresponde al día 20, que es la posición 19 del vector). En el caso del eje Y, el valor de dicho punto dependerá del precio de cada acción en el día concreto. Como se ha mencionado anteriormente, las imágenes que recibe el modelo no incluyen la representación de los ejes, por lo que se normalizarán los datos.

La manera de normalizar los datos será la siguiente: en primer lugar, se discretizará el eje X, cambiando las fechas correspondientes a los días representados por los número del 0 al 19; de este modo bastará con dividir el valor de la coordenada X en el punto de corte entre 20, para obtener su valor normalizado, indicando su posición en la imagen. En el caso del eje Y, al ser la escala variable, se normalizará la coordenada siguiendo la siguiente fórmula, teniendo en cuenta los valores mínimo y máximo de entre todas las curvas representados en la imagen:

$$y_{norm} = \frac{y_{interseccion} - minimo}{maximo - minimo} \quad (7)$$

Por ejemplo, para la Ilustración 14, el punto de intersección sería [18.34, 7.427889430231082], y una vez normalizado [0.917, 0.094]. Estas coordenadas normalizadas indican la posición del punto en términos de dónde se encuentra en la imagen, y fácilmente pueden ser traducidas a píxeles. Para las imágenes que recibe el modelo, cuya resolución es de 37x37 píxeles, el cruce se produciría en los píxeles [34, 3].

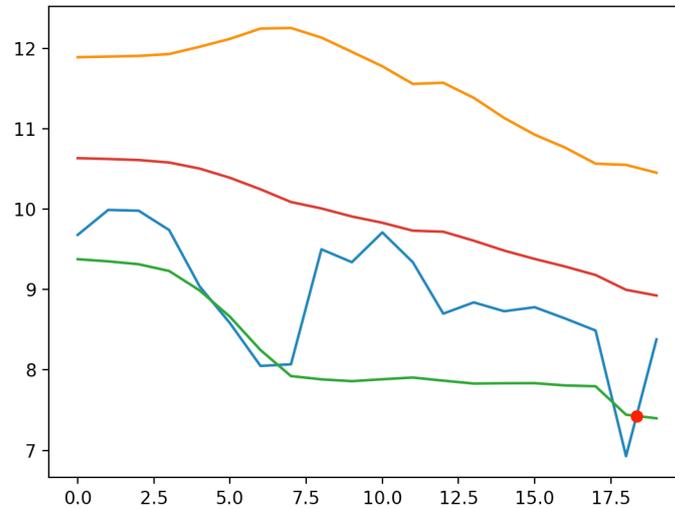


Ilustración 14. Representación del punto de corte para una señal de venta

Las imágenes que recibirá este modelo son las mismas que para los modelos anteriores, para los que se incluía la representación de las cuatro curvas. El cambio estará en los datos de salida, en este caso, será un vector de dos posiciones que contendrá las coordenadas normalizadas del punto de intersección para cada muestra. Para las imágenes en las que no se produce la intersección, y, por lo tanto, no presentan señal de ningún tipo, se fijará este dato como [0,0].

Se desarrollarán dos modelos diferentes, uno para cada clase de señales, es decir, un modelo con los datos de entrenamiento únicamente pertenecientes a imágenes con señal de venta, y lo mismo para las señales de compra. En primer lugar, se clasificarán las imágenes con el modelo de clasificación entrenado previamente, y las imágenes pasarán posteriormente a un modelo de regresión u otro según la clase predicha. Por ejemplo, si una imagen ha sido clasificada como señal de venta, el punto de cruce será predicho por el modelo de regresión entrenado con los datos pertenecientes únicamente a muestras que presentan señal de venta, independientemente de si la clasificación es correcta o no.

En cuanto a la estructura de los modelos, la principal diferencia respecto a los modelos anteriores se encuentra en la última capa, ya que en este caso, el objetivo no es clasificar sino realizar una regresión. Al ser más compleja esta tarea, ya que se busca la máxima precisión buscando el punto exacto de corte, el modelo contará con más de una capa densamente conectada. En primer lugar, una o más capas cuya activación será lineal, que dará lugar a la función de regresión buscada. La última capa tendrá una activación de tipo sigmoide para

asegurar que el valor predicho se encuentre entre 0 y 1 en todos los casos, ajustándose así a las coordenadas normalizadas.

Otro cambio importante en el modelo, será el cambio de la función de coste. En la clasificación, el dato más relevante a la hora de comparar los resultados de los modelos era la precisión. En este caso, lo más importante será como de cerca está el punto predicho del punto real, por lo tanto, al tratarse de una regresión, la función de coste pasará a ser el error cuadrático medio. El error cuadrático medio (mse) se calcula según la siguiente ecuación:

$$MSE(Y_i) = \frac{1}{Num_{muestras}} * \sum (Y - \hat{Y})^2 \quad (8)$$

Finalmente, la arquitectura del modelo convolucional que mejores resultados consigue para ambos tipos de señales es la siguiente:

- Misma estructura en las capas convolucionales que los modelos de clasificación
- Una capa densamente conectada con activación lineal de 50 unidades con dropout del 20%
- Una capa densamente conectada con función de activación sigmoide
- Función de coste: Error cuadrático medio
- 5 particiones para validación cruzada
- 50 épocas

Con esta arquitectura, se obtiene el error cuadrático medio para ambos modelos (compra y venta). Los resultados se muestran en la Tabla 9. Resultados del error cuadrático medio de los modelos convolucionales de regresión

MODELO	MSE
COMPRA	0,00097
VENTA	0,00272

Tabla 9. Resultados del error cuadrático medio de los modelos convolucionales de regresión

La Ilustración 15 e Ilustración 16 muestran dos ejemplos de la predicción del punto de cruce respecto al punto real. En el caso del ejemplo de muestra con señal de compra, el punto real está representado de color azul oscuro, y la predicción de color claro. En el caso de la muestra con señal de venta, el punto real está representado de color rojo, y el predicho de morado. Estas imágenes se han representado incluyendo los días posteriores al día correspondiente al punto de cruce para una mejor visualización.

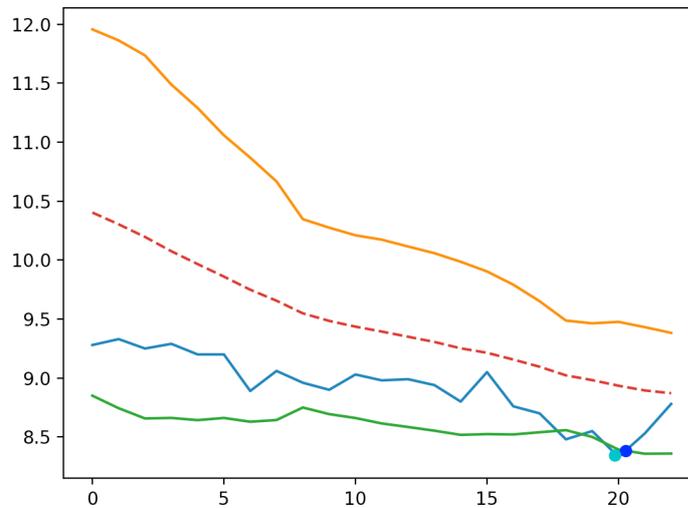


Ilustración 15. Comparación de la predicción del punto de corte respecto al punto real para una muestra con señal de compra

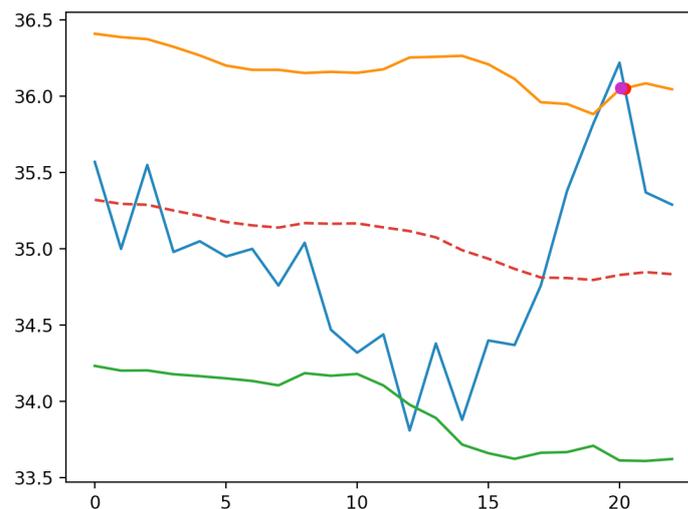


Ilustración 16. Comparación de la predicción del punto de corte respecto al punto real para una muestra con señal de venta

La Ilustración 17 muestra un ejemplo de predicción para una muestra que no presenta señal. En este caso, al no producirse ningún corte en el día de la curva del precio con una de las bandas, el valor de este punto está fijado en la coordenada [0,0].

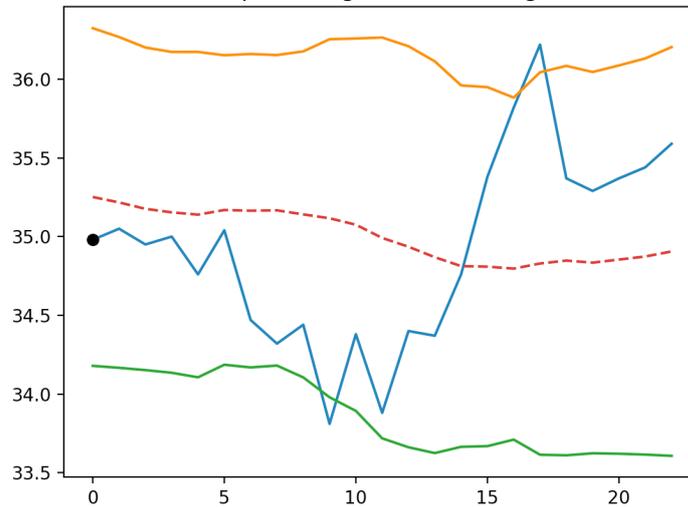


Ilustración 17. Comparación de la predicción del punto de corte respecto al punto real para una muestra sin señal

En esta aplicación en la que los datos son diarios, este punto de corte no representa un momento concreto de un día, ya que la curva del precio se representa uniendo al valor del precio de un día y del día siguiente, por lo que en esta recta se pierde la información de lo que pasa entre medias. Que el cruce se produzca más a la derecha en la imagen no implica necesariamente que el momento idóneo para operar sea la hora equivalente de este día. Es por esto que sólo se ha realizado la regresión para uno de los modelos de clasificación detallados anteriormente.

Esta técnica tendría más sentido aplicarla para datos intradiarios y así aplicarla a una estrategia de inversión de alta frecuencia. El precio de una acción en un mismo día puede variar significativamente, por lo que únicamente con el valor de cierre se pierde mucha información. Sin embargo, trabajando con datos intradiarios en intervalos de minutos o incluso segundos, el punto de corte representaría de forma más precisa el momento exacto en el que realizar una operación de compra o venta, ya que en este caso no se perdería información tan relevante.

3.3. Modelos de machine learning

En el apartado anterior se han analizado los resultados de entrenar modelos de aprendizaje profundo con imágenes para resolver un problema financiero, en concreto, identificar oportunidades de compra o venta de una acción según un algoritmo financiero algebraico. Como se puede concluir de los resultados mostrados en las diferentes tablas que contienen la precisión de distintos modelos al realizar esta tarea, el uso de estos modelos demuestra ser beneficioso para conseguir recuperar las distintas señales de compraventa.

Estos modelos, al contar con numerosas capas y al ser los datos imágenes, tienen muchos más parámetros que un modelo clásico numérico, por lo que tardan más tiempo en ser entrenados y tienen un coste computacional muy alto. Por esto, aunque presenten buenos resultados, resulta conveniente comprarlos con modelos más sencillos, que cuentan con menos parámetros y para los cuales el coste computacional es mucho menor. Una vez demostrada la capacidad de los modelos convolucionales, comparándolos con modelos más sencillos, se analizará si estos últimos modelos son capaces de obtener resultados similares, necesitando menos parámetros y un tiempo de ejecución menor, para estudiar la utilidad de aumentar la complejidad de los modelos a entrenar para realizar la tarea en cuestión.

A continuación, se entrenarán tres modelos diferentes con datos de entrada numéricos, clásicos de machine learning. Estos modelos recibirán como datos de entrada un vector conteniendo el valor del precio de cierre de cada acción durante 20 días, equivalente a los días representados en las imágenes de los modelos convolucionales.

Los modelos entrenados serán los siguientes: regresión lineal, support vector machine y random forest.

Todos los modelos se entrenarán con validación cruzada de diez particiones. Por otro lado, se modificará el formato de los datos de salida. Por la naturaleza de la estructura de estos modelos, en lugar de un vector de tres posiciones, la salida será representada con un único número, que tomará un valor entre tres posibles, uno para cada clase. Este número es equivalente a la posición en el vector que en los modelos anteriores contenía un 1: será 0 para las muestras sin señal, 1 para las muestras con señal de compra y 2 para las muestras con señal de venta.

3.3.1. Modelo de regresión lineal

El primer modelo entrenado será un modelo de regresión lineal. Con este modelo se busca comprobar si con una función tan simple como la regresión lineal se pueden reproducir o mejorar los resultados obtenidos con las redes neuronales convolucionales. En la Tabla 10 se muestra el resultado de la precisión para este modelo.

ACCURACY			
Total	No Signal	Buy	Sell
0,12	0,1	0,933	0,066

Tabla 10. Resultados de precisión del modelo de regresión lineal

De estos resultados se puede concluir que una regresión lineal es demasiado simple para poder clasificar correctamente los datos. Los resultados de este modelo son comparables con los obtenidos por los modelos convolucionales para los cuales los datos recibidos sólo incluían la representación de la curva del precio, ya que para ambos la información recibida será la misma. La precisión baja de un 85,5% a un 12%.

Los resultados en la precisión son tan bajos porque la mayor parte de las muestras han sido clasificadas como compra, como se puede comprobar en la Tabla 11.

		NO		
		SEÑAL	COMPRA	VENTA
SEÑAL REAL	PREDICCIÓN			
	NO SEÑAL	403	3615	12
	COMPRA	7	98	0
	VENTA	0	71	5

Tabla 11. Comparación de las predicciones frente a los valores reales de las muestras para el modelo de regresión lineal

3.3.2. Modelo de support vector machine

Las máquinas vector de soporte son un algoritmo simple igual que la regresión lineal, pero algo más complejas en comparación. Son utilizadas para tareas tanto de clasificación como de regresión, más comúnmente para clasificación [19]. El objetivo de este algoritmo es encontrar un hiperplano que separe los datos por clases, de dimensión igual al número de elementos de los datos de entrada, en este caso, 20.

Para una misma aplicación puede haber muchos hiperplanos diferentes capaces de separar los datos según las diferentes clases. El objetivo de este algoritmo es encontrar el plano que tenga el máximo margen, es decir, la máxima distancia entre puntos de cada clase.

En la Ilustración 18 se muestra un ejemplo con diferentes márgenes para un set de datos con dos clases. Ambas opciones separan correctamente los clases, cambiando la orientación y la distancia a las muestras. El margen escogido por el algoritmo en este caso, sería el de la imagen de la derecha, es decir, el mayor de los dos. Los vectores de apoyo son los datos (las muestras) que se encuentran más cerca del hiperplano e influyen tanto la posición como la orientación de este. Son estos puntos los que el algoritmo utiliza para maximizar el margen.

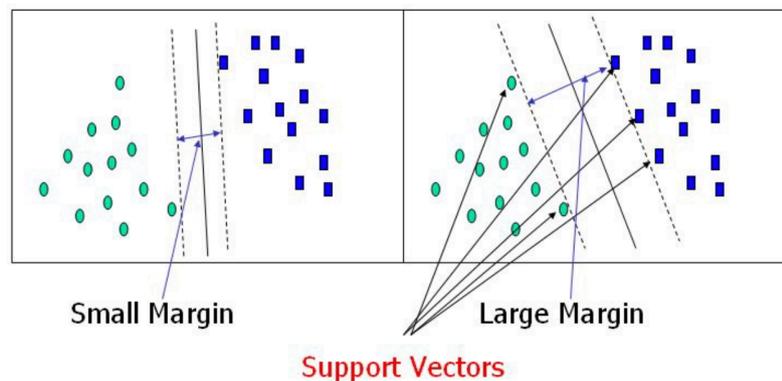


Ilustración 18. Ejemplo de diferentes márgenes para los mismos datos

La Tabla 12 y la Tabla 13 muestran los resultados obtenidos con este modelo. La precisión en este caso es del 50,8%, significativamente mayor que para el modelo de regresión lineal al ser este un algoritmo más complejo, y por tanto capaz de identificar mejor los factores que distinguen una clase de otra. Comparado con los modelos neuronales convolucionales entrenados, ocurre lo mismo que con la regresión lineal, la precisión es mucho más baja, un

50,8% respecto a un 85,5%. Aunque con una máquina de vector de soportes se consiguen mejores resultados que con una simple regresión lineal, este algoritmo sigue sin ser capaz de realizar la tarea de clasificación adecuadamente.

ACCURACY			
Total	No Signal	Buy	Sell
0,508	0,487	0,981	0,961

Tabla 12. Resultados de precisión del modelo SVM

El principal fallo de este algoritmo es la clasificación de las imágenes que no presentan ninguna señal, clasificando menos del 50% correctamente. Sin embargo, las muestras tanto de venta como de compra tienen una precisión cercana al 100%.

		NO		
		SEÑAL	COMPRA	VENTA
PREDICCIÓN				
SEÑAL REAL	NO SEÑAL	1963	1374	693
	COMPRA	2	103	0
	VENTA	3	0	73

Tabla 13. Comparación de las predicciones frente a los valores reales de las muestras para el modelo de SVM

3.3.3. Modelo random forest

Los algoritmos de random forest consisten en una combinación de varios árboles de decisión individuales [20]. Cada árbol de decisión predice una clase como resultado, la clase más veces predicha por la suma de todos los árboles será la elegida por el modelo de random forest. El motivo de utilizar random forest y no simples árboles de decisión se debe a que un gran número de modelos no correlados operando en conjunto superarán en rendimiento a cualquiera de los árboles individuales. El modelo entrenado cuenta con 6 árboles de decisión.

Los resultados de este modelo, representados en la Tabla 14 y la Tabla 15 superan los resultados obtenidos con el modelo de regresión lineal, pero no se alcanza la precisión conseguida con el modelo de support vector machine, y por lo tanto, la precisión resultante se encuentra muy por debajo de la conseguida por los modelos neuronales convolucionales. La precisión total en este caso es de 28,8 %.

ACCURACY			
Total	No Signal	Buy	Sell
0,288	0,267	0,867	0,618

Tabla 14. Resultado de precisión del modelo Random Forest

Para este modelo, al igual que para los demás, la precisión para las muestras sin señal es muy baja, prediciendo la mayor parte de las imágenes erróneamente.

		PREDICCIÓN	NO		
			SEÑAL	COMPRA	VENTA
SEÑAL REAL	NO SEÑAL	1074	1967	989	
	COMPRA	14	91	0	
	VENTA	13	16	47	

Tabla 15. Comparación de las predicciones frente a los valores reales de las muestras para el modelo de Random Forest

4. Análisis de los resultados

En total se han entrenado cinco redes neuronales convolucionales, según su objetivo y el tipo de datos que ha recibido cada una.

Los cuatro primeros modelos tenían como objetivo realizar una tarea de clasificación, con la finalidad de identificar las muestras para las cuales se presenta una oportunidad de compra o venta del activo financiero al que pertenece esa muestra. Esta clasificación se ha realizado para dos situaciones diferentes, en primer lugar, para identificar dichas oportunidades en el último día representado, asumiendo que se conoce de antemano la información del día para el cual se quiere saber si la acción a tomar debería ser de compra o venta, o ninguna. En segundo lugar, la clasificación se ha realizado a modo de predicción, con el fin de identificar la señal correspondiente al día siguiente al último representado, sin conocer información del precio de cierre de la acción de ese día.

A parte, para ambos modelos, se ha realizado una modificación en los datos de entrada, incluyendo únicamente la representación de la curva de la evolución del precio de cierre de la acción, eliminando de las imágenes las curvas correspondientes al algoritmo de las Bandas de Bollinger.

En la Tabla 16 se muestra la precisión alcanzada para cada uno de estos cuatro modelos. La precisión indica el porcentaje de las muestras que han sido clasificadas correctamente. Todos los modelos, excepto el modelo de predicción cuyos datos de entrada incluían la representación de todas las curvas alcanzan una precisión del 85% aproximadamente. De todos los modelos, el que mayor precisión ha obtenido es el modelo de predicción que únicamente incluye la representación del precio de cierre. Este último modelo es el más interesante de todos ya que es el que menos información contiene, siendo por esto el más útil y práctico a la hora de ser utilizado como herramienta de inversión. El tercer modelo, que realiza una predicción a partir de las imágenes que incluyen la representación de todas las curvas, a penas alcanza una precisión del 53,5%. Este resultado, una precisión un 32% menor que el último modelo que realiza la misma tarea de predicción únicamente con la representación de la curva del precio, puede tener como consecuencia que las imágenes contienen demasiada información que puede confundir al modelo, clasificando incorrectamente prácticamente la mitad de las muestras.

Para los dos tipos de modelo, clasificación y predicción, la precisión obtenida es mayor para los datos que únicamente incluyen la representación de precio de cierre, siendo la diferencia de a penas un 0,8% en el caso de la clasificación, y de un 32% en la predicción. En el caso de los modelos que incluyen la representación de todas las curvas, son necesarias más épocas en el entrenamiento de las establecidas en la estructura base, entre 100 y 150. Para los modelos entrenados con las imágenes eliminando la representación de las bandas de Bollinger, con el número de épocas de la arquitectura base, 50, es como se consiguen los mejores resultados. Para los cuatro modelos la precisión más alta se obtiene para una validación cruzada con 10 particiones.

MODELO	PRECISIÓN
Clasificación, representación de todas la curvas	84,5%
Clasificación, únicamente curva del precio	85,3%
Predicción, representación de todas las curvas	53,5%
Predicción, únicamente la curva del precio	85,5%

Tabla 16. Resultados de precisión finales de todos los modelos convolucionales

Por último, se ha entrenado una quinta red neuronal convolucional, esta vez para realizar una tarea de regresión. Este modelo busca identificar el punto de corte de la curva del precio de cierre con la banda superior para las muestras con señal de venta, y el corte con la banda inferior para las muestras con señal de compra. Se han entrenado dos modelos de este tipo, uno para cada tipo de señal. Los resultados de estos modelos se analizan buscando minimizar el error cuadrático medio, en lugar de aumentar la precisión. Los resultados obtenidos se muestran en la Tabla 17 se recogen los resultados para estos modelos.

MODELO	MSE
COMPRA	0,00097
VENTA	0,00272

Tabla 17. Resultados del error cuadrático medio de los modelos convolucionales de venta y compra

A parte de los diferentes modelos neuronales convolucionales, propios de Deep learning, se han entrenado tres modelos diferentes de machine learning: una regresión lineal, una máquina vector de soporte y un random forest. Estos modelos toman como datos de entrada un vector de 20 posiciones, con la información del valor del precio de cierre de cada acción durante 20 días. Estos modelos son entrenados con la misma información que el modelo neuronal convolucional de clasificación para el cual los datos de entrada eran imágenes con la representación del precio de cierre únicamente. La diferencia entre la información que reciben los modelos de machine learning respecto al modelo convolucional es que para los primeros el formato es numérico, mientras que para el segundo son imágenes. Por esto, los resultados de los modelos de machine learning se compararán con este modelo convolucional de los diferentes entrenados.

Los resultados de la precisión de estos modelos se muestran en la Tabla 18. Como se puede observar, la red neuronal convolucional presenta una precisión mayor que la de todos los modelos de machine learning, siendo entre un 34,5% y un 73,3% mayor.

MODELO	PRECISIÓN
Regresión lineal	12%
Support Vector Machine	50,8%
Random forest	28,8%
Red neuronal convolucional	85,3%

Tabla 18. Comparación de los resultados de los modelos de machine learning frente a una red neuronal convolucional

De la Tabla 17 y la Tabla 18 se pueden extraer dos conclusiones principales. En primer lugar, los modelos neuronales convolucionales, que toman imágenes como datos de entrada, para tres de los cuatro modelos diferentes entrenados, presentan una precisión alrededor del 85%. Los modelos son capaces de recuperar con éxito la mayor parte de las señales de compra y de venta, distinguiendo a su vez los días en los que no se presenta ninguna de estas señales. Con estos resultados, se puede confirmar que al transformar el planteamiento clásico de este problema financiero centrado en la compraventa de activos de un problema numérico a uno visual se consigue realizar la tarea de clasificación con éxito.

Por otro lado, comparando con los resultados obtenidos por los modelos de machine learning, se confirma que no sólo son los modelos neuronales convolucionales capaces de resolver este problema, si no que superan con creces los resultados de estos modelos más simples. Los modelos de machine learning demuestran no ser lo suficientemente complejos para clasificar las muestras correctamente, por lo que resulta beneficioso emplear modelos neuronales convolucionales. Estos modelos podrán obtener resultados de precisión similares a los de las redes neuronales haciendo una ingeniería de variables adecuada. En este caso han sido entrenados con una única variable, el precio de cierre de la acción. Añadir otras variables como el precio de apertura, o el máximo diarios podría aumentar la precisión alcanzada. Una de las ventajas de las redes neuronales convolucionales es que no es necesario realizar esta ingeniería de variables, ya que al trabajar con imágenes, las redes son capaces de obtener un mayor nivel de abstracción de la información.

5. Conclusiones y futuros desarrollos

Una vez analizados los resultados de los diferentes modelos entrenados, tanto las redes neuronales convolucionales como los modelos de machine learning, se puede concluir que el uso de los modelos convolucionales demuestra ser beneficioso para conseguir recuperar las distintas oportunidades de compra y venta de activos financieros.

Se han resuelto con éxito las ampliaciones propuestas, demostrando la versatilidad tanto de las redes neuronales convolucionales como del nuevo enfoque centrado en las imágenes como datos de entrada. Si bien la segunda red desarrollada (en concreto el modelo de predicción con información de las curvas tanto del precio como del algoritmo) a penas alcanza una precisión del 53,5%, el último modelo, que realiza la misma tarea con la misma finalidad pero con menos información de entrada, alcanza la mayor precisión de todos los modelos, del 85,5%. Este último modelo resulta ser el más práctico ya que es el que se ajusta más a una aplicación real, en la que se cuenta con la menor información posible.

Por último, la comparación con los resultados obtenidos para los modelos de machine learning, cuyos datos de entrada eran numéricos, permite comprobar que no solo son capaces las redes convolucionales de resolver las tareas propuestas en este trabajo, si no que consiguen alcanzar una precisión mucho mayor, confirmando la utilidad de utilizar modelos de aprendizaje profundo para aplicaciones complejas.

En este proyecto se ha trabajado con datos diarios. Los modelos convolucionales entrenados realizan una tarea de clasificación para días concretos, unos para un día pasado para el cual se conoce el precio de cierre y otros para el día siguiente, sin conocer información a cerca del precio para ese día. Este último modelo es el que tiene más sentido seguir desarrollando, ya que en la realidad, no se conoce de antemano el precio que va a alcanzar un activo en el futuro.

Esta aplicación podría ser más útil aplicada a datos intradiarios, en los que la ventana de tiempo con la que se trabaja sea de 20 minutos o incluso segundos, en lugar de días. De esta forma, el modelo convolucional de regresión entrenado podría ser aplicado para calcular el modelo

exacto del día en el que se debería comprar o vender una acción, para seguir una estrategia de inversión de alta frecuencia.

Por otro lado, como futuro desarrollo adicional, se propone aumentar el número de muestras con las que se entrenen los modelos para así conseguir una mayor precisión en la clasificación.

Referencias

- [1] Investopedia. [En línea]. Available: <https://www.investopedia.com/terms/f/financial-market.asp> https://en.wikipedia.org/wiki/Financiacional_market <https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/financial-markets/> .
- [2] F. Institute. [En línea]. Available: <https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/financial-markets/>.
- [3] [En línea]. Available: <https://polygant.net/ai/artificial-intelligence-for-trading>.
- [4] [En línea]. Available: <https://readwrite.com/2019/12/23/5-ways-artificial-intelligence-technology-is-transforming-the-financial-markets/>.
- [5] [En línea]. Available: <https://builtin.com/artificial-intelligence/ai-trading-stock-market-tech>.
- [6] J. Capital. [En línea]. Available: <https://www.j4capital.trade/>.
- [7] «Greenkey Technologies,» [En línea]. Available: <https://greenkeytech.com/home/>.
- [8] N. Cohen, T. Balch y M. Veloso, «Trading via Image Classification,» *JP Morgan*, 2019.
- [9] U. Nations, «Objetivos de desarrollo sostenible,» [En línea]. Available: <https://www.un.org/sustainabledevelopment/es/infrastructure/>.
- [10] Wikipedia, «Deep Learning,» [En línea]. Available: https://en.wikipedia.org/wiki/Deep_learning.
- [11] M. L. Mastery, «What is deep learning,» [En línea]. Available: <https://machinelearningmastery.com/what-is-deep-learning/>.
- [12] Researchgate. [En línea]. Available: https://www.researchgate.net/figure/The-performance-of-deep-learning-with-respect-to-the-amount-of-data_fig3_331540139.
- [13] T. d. science, «A comprehensive guide to convolutional neural networks,» [En línea]. Available: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>.
- [14] MDPI. [En línea]. Available: <https://www.mdpi.com/2076-3417/9/21/4500>.
- [15] MC.AI. [En línea]. Available: <https://mc.ai/convolution-operation-comprehensive-guide/>.

- [16] [En línea]. Available: <https://ailephant.com/glossary/relu-function/>.
- [17] [En línea]. Available: <https://github.com/alvarobartt/investpy> .
- [18] «Investopedia,» [En línea]. Available:
<https://www.investopedia.com/terms/b/bollingerbands.asp>.
- [19] [En línea]. Available: <https://decodingmarkets.com/mean-reversion-trading-strategy/#:~:text=A%20mean%20reversion%20trading%20strategy%20involves%20betting%20that%20prices%20will,develop%20strategies%20for%20both%20phases..>
- [20] «Support Vector Machine Introduction,» [En línea]. Available:
<https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>.
- [21] «Understanding Random Forest,» [En línea]. Available:
<https://towardsdatascience.com/understanding-random-forest-58381e0602d2#:~:text=The%20random%20forest%20is%20a,that%20of%20any%20individual%20tree..>
- [22] Polygian, «Artificial intelligence for trading,» [En línea]. Available:
<https://polygant.net/ai/artificial-intelligence-for-trading/>.