

Predicción del precio del bitcoin mediante redes neuronales recurrentes

Daniel Ramos Hoogwout

Tutor: *Victor Troster*

Treball de fi de Màster Universitari en Anàlisi de Dades Massives en Economia i Empresa
(MADM)

Universitat de les Illes Balears^[1]
07122 Palma de Mallorca
dramoshoogwout@gmail.com

Resumen

En los últimos años se ha incrementado exponencialmente la utilización de criptomonedas, una moneda virtual que pretende sustituir al dinero fiduciario emitido por los bancos centrales de los países. Este tipo de activos ha crecido tanto en número como en capitalización al nivel de poder compararse incluso con índices bursátiles, por ello el objetivo de este estudio es intentar predecir el precio del Bitcoin, la criptomoneda por excelencia, mediante un tipo de redes neuronales recurrentes, en concreto, *Long Short-Term Memory* (LSTM), uno de los algoritmos de *deep learning* más avanzados que existen que gracias a su celda de memoria permiten almacenar información del pasado para hacer predicciones más precisas.

Abstract

In recent years, the use of cryptocurrencies has increased exponentially, a virtual currency that aims to replace fiduciary money issued by the central banks of countries. This type of asset has grown both in number and in capitalization to the level of being able to be compared even with stock market indexes, for this reason the objective of this study is to try to predict the price of Bitcoin, the cryptocurrency by excellence, through a type of recurrent neural network, specifically, *Long Short-Term Memory* (LSTM), one of the most advanced deep learning algorithms to date, that thanks to its memory cell, allows to store past information to make more precise predictions.

Palabras clave: Predicción, Series Temporales, Redes Neuronales, LSTM, Bitcoin, Cripto.

1. Introducción

Las redes neuronales recurrentes representan uno de los algoritmos más avanzados que existen en el mundo del *deep learning* supervisado.

Como ya su propio nombre indica una red neuronal intenta imitar un cerebro humano para resolver problemas, en este caso, la red neuronal recurrente intenta imitar el lóbulo frontal, la parte del cerebro humano que se encarga de la memoria a corto plazo. Haciendo uso de esta memoria el algoritmo es capaz de reaccionar a situaciones futuras con información del pasado y por este motivo hace que este tipo de algoritmos sean tan potentes para predecir series temporales.

Las redes neuronales recurrentes, en concreto, la variante de *Long Short-Term Memory* (LSTM) son redes neuronales muy adecuadas para datos de series temporales ya que gracias a su estructura incluye una celda de memoria que permite mantener información por periodos prolongados en el tiempo venciendo así al problema del gradiente descendiente.

Este tipo de redes se han aplicado con resultados notables en aplicaciones como reconocimiento de voz, lenguaje natural, síntesis de voz... Por ello se va a utilizar este tipo de redes neuronales en este trabajo para intentar predecir el precio del bitcoin

haciendo uso de datos de series temporales con su precio pasado entre otros indicadores para poder hacer una predicción lo más precisa posible

1.1. Redes Neuronales Recurrentes

Una red neuronal recurrente es un tipo de red neuronal que dentro de su estructura puede retroalimentarse, es decir, utiliza su propio *feedback* para influenciar la entrada actual y la salida.

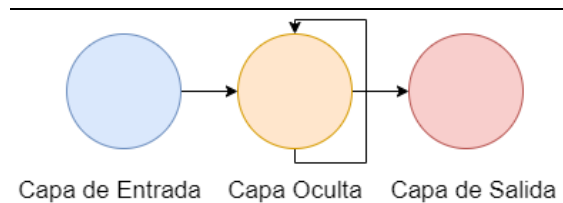


Figura 1: Diagrama de la red neuronal recurrente

Por tanto, la salida de una red neuronal recurrente depende de los elementos previos de la propia secuencia. Esto genera una característica distintiva, la red neuronal recurrente comparte el mismo parámetro de peso dentro de cada capa de la red.[1]

1.2. Desvanecimiento del gradiente

Este problema que presentan las redes neuronales recurrentes fue descubierto por Josep Hochreiter.[2]

El problema radica que en la red neuronal utiliza un algoritmo de gradiente descendiente para encontrar el mínimo global de la función de costes que es la configuración óptima de la red.

En el caso de las redes neuronales recurrentes la información pasada se utiliza como input para determinar la predicción presente por lo tanto cuando se actualizan los pesos para minimizar el error se tendrá que actualizar todas las neuronas que han participado en el proceso.

Al principio del proceso se asignan pesos aleatorios con valores cercanos a cero y entonces se comienza a entrenar la red, al multiplicar todos los momentos pasados por un valor cercano a cero el gradiente se reduce tras cada multiplicación por lo que al final se hace cada vez más difícil actualizar los pesos de la red y aumenta el tiempo en el que se obtiene el resultado final.

1.3. LSTM

Es una variación de la red neuronal recurrente que soluciona el problema comentado en la sección anterior. Esta evolución añade celdas de entrada, salida y olvido.

[Tengo que ver el video](#)

2. Metodología

En esta sección se expone la metodología utilizada para obtener el modelo final expuesto en la siguiente sección.

Se comienza con un modelo sencillo

2.1. Construcción de la red neuronal

2.2. Robustez

2.3. Modelo inicial

2.4. Hiperparametrización

2.5. Modelo final

2.6. Multistep forecasting

3. Análisis Empírico

En esta sección se aplican las redes neuronales recurrentes explicadas en la sección anterior para predecir el precio de apertura del Bitcoin durante un período de tres meses.

El modelo ha ido evolucionando desde un modelo inicial donde solamente se tenía en cuenta el precio de apertura del bitcoin para utilizarlo como modelo de partida e ir ajustando y añadiendo información hasta llegar al modelo que presenta más adelante.

Para la muestra de entrenamiento se ha escogido el período que abarca desde 17 de septiembre de 2015 hasta el 30 de noviembre de 2019. Se ha optado por este período ya que comienza desde el dato más antiguo disponible en Yahoo Finance hasta tres meses antes de que el efecto de la pandemia fuese global, considerando marzo como el inicio de la pandemia de COVID-19. Por tanto, la muestra de testeo comienza desde el 1 de diciembre de 2019 hasta el 29 de febrero de 2020.

Finalmente se ha escogido como muestra de validación el periodo que abarca desde el 1 de marzo de 2021 hasta el 27 de noviembre de 2021, se utiliza para dar una estimación imparcial de la habilidad predictora del modelo.

Yahoo Finance ofrece varios datos referentes a la cotización de activos de los cuales se han escogido el precio de apertura y de cierre diarios para entre-

nar la red neuronal. En un principio también se había incluido un índice bursátil, en concreto, el IBEX 35 ya que en los últimos meses se ha comentado mucho la correlación creciente entre el bitcoin y los índices bursátiles, desafortunadamente la inclusión del índice empeoró mucho las predicciones y se ha optado por eliminarlo de este análisis.

La entrada de información a la red neuronal está compuesta por una matriz tridimensional que contiene el precio de apertura y de cierre del bitcoin con 90 *timesteps*, es decir, en cada momento del tiempo la red neuronal será capaz de observar 90 días previos al día actual. Previamente se había hecho uso de 60 días como período base, pero el incremento a 3 meses supuso una mejora considerable en el modelo.

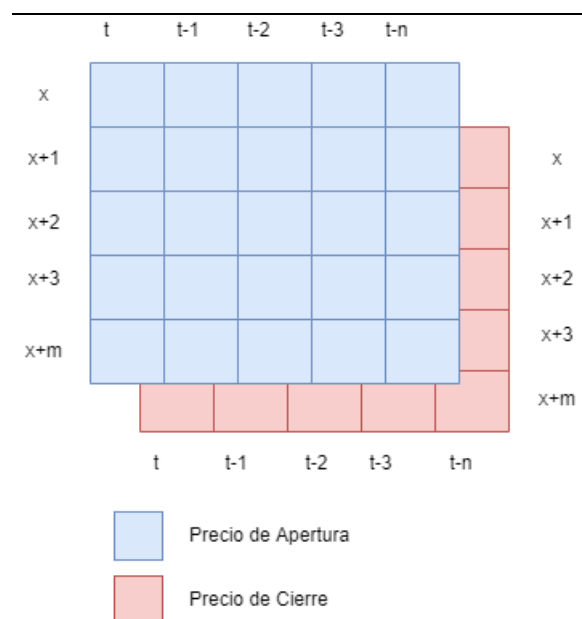


Figura 1: Matriz de Entrada

Antes de introducir estos datos en la red neuronal estos se han normalizado a valores entre 0 y 1 para simplificar la entrada a la red neuronal, evitando así que una variable domine sobre otra en el caso de que sus magnitudes fuesen diferentes.

La red neuronal que procesa esta matriz está compuesta por una entrada, cuatro capas con cincuenta neuronas cada una intercaladas por una capa de dropout que permite que se desactiven un veinte por ciento de las neuronas, por tanto, solo el ochenta por ciento de la información pasará a la siguiente capa previniendo así el sobreajuste. Finalmente hay una capa de salida que proporciona la predicción del precio de apertura del bitcoin.

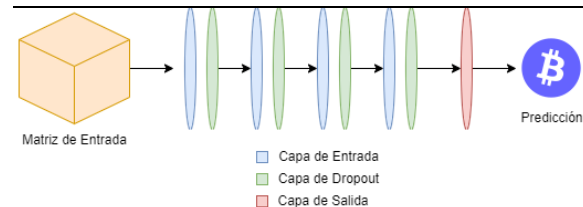


Figura 2: Estructura de la red neuronal

Se ha optado por alimentar la red neuronal en *batches* de 32 unidades ya que es un valor relativamente pequeño que ayuda a converger de manera rápida a costa de introducir ruido en el proceso de entrenamiento.

En un principio el número de *epochs* o en otras palabras el número de veces que el algoritmo de aprendizaje ha digerido todo el *dataset* de entrenamiento se había situado en 100. Utilizando el parámetro de *Early Stopping* del paquete de *keras* se ha utilizado una paciencia de 10 *epochs* para monitorizar si el modelo dejaba de minimizar el gradiente descendiente en las siguientes 10 iteraciones ya que al usar *batches* de un tamaño de 32 unidades el modelo aprende con mucha rapidez y puede ocasionar *overfitting*.

En este modelo, el *Early Stopping* se activa a los 46 *epochs*, es decir, desde el *epoch* 36 no hay una disminución de la pérdida del gradiente descendiente del modelo y por tanto selecciona los pesos del *epoch* 36.

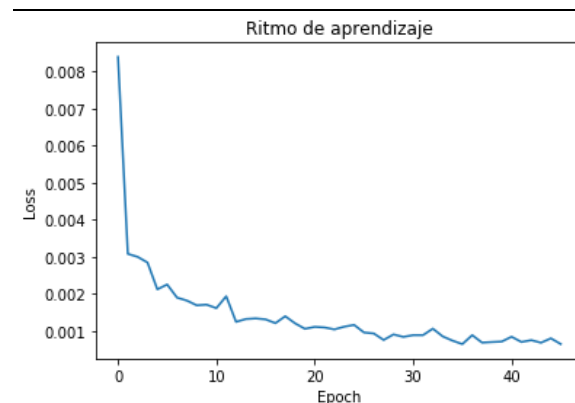


Figura 3: Ritmo de aprendizaje de la RNR

Una vez realizado el entrenamiento del modelo se predicen los siguientes 3 meses con la muestra de test y se obtiene el siguiente gráfico.



Figura 4: Predicción sobre la muestra de test

Para poder comparar esta predicción con otros modelos se estima la raíz del error cuadrático medio, así como también el error medio absoluto.

Finalmente se utiliza este modelo para predecir el precio con una muestra de validación utilizando un período futuro, en concreto, comienza a predecir junio de 2021 pudiendo así coger los 90 días anteriores para hacer una predicción.

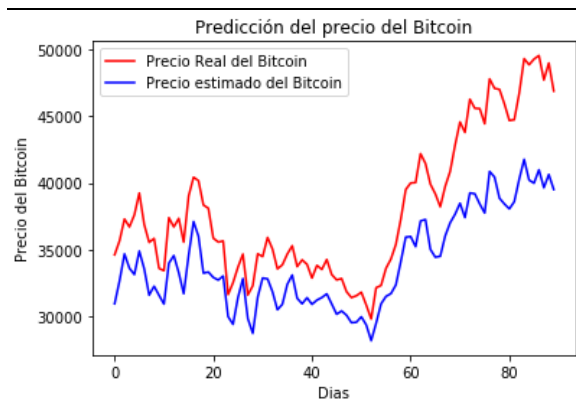


Figura 5: Predicción sobre la muestra de validación

En este caso también se obtiene la raíz del error cuadrático medio y el error medio absoluto para así poder comparar los modelos con los resultados de la muestra de test y determinar si ha habido sobreajuste.

Falta una tabla con los resultados de varios modelos

4. Conclusiones

Falta rellenar esta sección.

-El problema del gradiente para series temporales largas se puede resolver añadiendo más capas a la red.

5. Bibliografía

Apéndice A. Configuración de Anaconda

Es recomendable el uso de apéndices cuando se pretende aclarar una parte del trabajo paralela a la principal línea argumental, por ejemplo: la configuración de una determinada herramienta, la puesta a punto de un entorno, los resultados secundarios de un experimento, etc.

Apéndice B. Modelos descartados

En la documentación no es recomendable el uso excesivo de apéndices.

Referencias