

# Proyecto de Redes Neuronales.

## Redes neuronales aplicadas a la predicción del precio de una acción en el mercado bursátil.

Franco Cerino

### Abstract

Se utilizaron diferentes modelos de redes neuronales recurrentes con la finalidad de evaluar su poder predictivo al momento de predecir el precio de las acciones de Tesla en un día, utilizando datos del precio de la acción y del volumen de operaciones de la misma.

También se ha utilizado un modelo simple, dado por una predicción constante, para obtener un valor de referencia y determinar efectivamente si el poder predictivo de las redes neuronales utilizadas es lo suficientemente bueno como para considerar a estos modelos como propensos para realizar predicciones certeras.

### 1. Introducción

A menudo los funcionarios de una empresa deciden poner en venta un porcentaje de la misma para obtener capital, en fracciones denominadas acciones.

Las acciones pueden ser adquiridas por cualquier individuo y se pueden transferir del propietario a otra persona a través de dinero. El precio de estas no es fijo, sino que su valor se regula a través de la oferta y demanda de dicho título (estas son las intenciones de venta y compra, respectivamente).

Usualmente se realiza operaciones con acciones con la expectativa de que estas aumenten o disminuyan su precio, para poder obtener un beneficio económico. Por ejemplo, en caso de obtener una acción a un precio dado y en el caso de que en el futuro esta valore en un porcentaje dado, el tenedor de la acción puede venderla a un ofertante para concretar una

ganancia económica.

Poder predecir el movimiento del precio de una acción no es una tarea sencilla, ya que este se ve afectado por un gran número de factores de diversa índole. Por esto se suele recurrir a diferentes enfoques para poder predecir el precio. Dentro de ellos, existe el llamado análisis técnico, que consiste en intentar predecir el precio solo a través de la tendencia de diferentes magnitudes de la empresa en el tiempo, donde principalmente se utilizan el precio de la acción y el volumen de la misma (cantidad de transacciones realizadas en un lapso determinado de tiempo).

Dentro de los métodos utilizados para predecir el precio de acciones, se ha comenzado a recurrir a modelos de Machine Learning, reconocidos por su gran poder predictivo en diferentes casos.

Cabe mencionar que la inferencia que se realiza al predecir el precio de una acción es una extrapolación de los datos, ya que se intenta predecir comportamientos fuera del dominio de donde se conoce información, es decir, que a través de datos del pasado se intenta conocer el comportamiento a futuro.

En este trabajo se utilizan redes neuronales recurrentes con la finalidad de realizar predicciones del precio de las acciones de Tesla en un día.

### 2. Metodología y Resultados

Se utilizó un dataset correspondiente a las acciones de Tesla, obtenido a través del paquete

yfinance, disponible para Python. Este otorga datos con frecuencia diaria.

Se utilizaron los datos del precio y volumen (cantidad de operaciones de compra-venta realizadas) de dicha acción, comprendidos en los días hábiles entre 6/9/2010 y 4/3/2021, siendo un total de 2659 días. Cabe mencionar que el valor de la acción utilizado corresponde al precio de cierre, es decir la última cotización del título en cada día. Esto se ha hecho para tomar valores del precio siempre en el mismo momento de cada día.

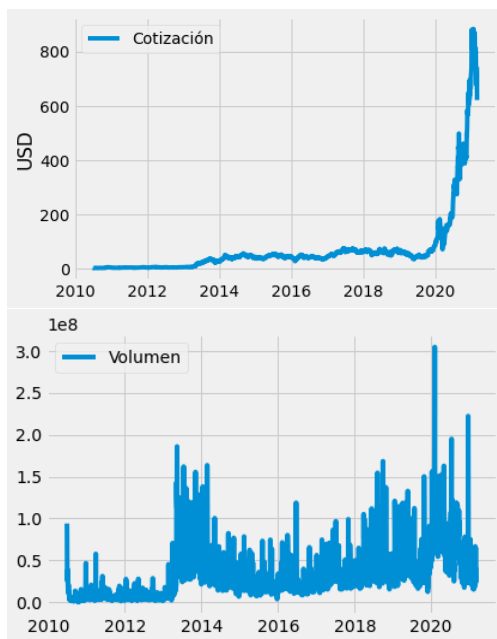


Figura 1: Precio y volumen de la acción de Tesla en el tiempo.

## 2.1. Preprocesamiento de datos

Antes de entrenar un modelo de Machine Learning puede ser necesario modificar el conjunto de datos inicial, por ejemplo para que pueda ser interpretado por el modelo. Incluso puede que un tratamiento mayor permita que el aprendizaje otorgue mejores resultados.

En este caso se contó con un conjunto de datos indexado en tiempo y sin valores faltantes. Se procedió normalizando los datos, formando pares input-output y dividiendo el dataset en

conjuntos de entrenamiento, validación y test, acordes a un problema de series temporales.

### Normalización de los datos

Una técnica muy utilizada es la de normalización, donde se varía la escala de todos los features al rango  $[0,1]$ . Esta es una buena práctica para trabajar con features comprendidos en la misma escala, que puede permitir mejorar la performance de los modelos utilizados.

### Disposición de los datos para un modelo de aprendizaje supervisado

Para llevar el conjunto de datos indexado en el tiempo a un formato de datos de un problema supervisado, es decir con valores de entrada y de salida, se formó pares de conjuntos de datos, donde de cada par, un elemento está dado por los datos de  $n$  días consecutivos (se suele denominar que se utiliza una ventana de  $n$  días), y el otro elemento está dado por el precio de la acción en el día subsiguiente a dicha secuencia.

Para un  $n$  dado, el primer par de datos se formó tomando los datos de los primeros  $n$  días como input y el precio del día  $n+1$  como output. Para el segundo par de datos, comenzando desde el segundo día, se tomó los datos de  $n$  días consecutivos como input y el dato del precio del día  $n+2$ , y así sucesivamente hasta formar todos los pares posibles.

De esta forma se puede buscar un mapa que tome datos de  $n$  días consecutivos y otorgue como resultado el precio de la acción en el día subsiguiente.

Al momento de entrenar el modelo no se sorteará los pares de datos aleatoriamente como se acostumbra, sino que se otorgarán en orden temporal.

### División en conjuntos de entrenamiento, validación y test

Para utilizar modelos de Machine Learning, se debe elegir conjuntos de datos para entrenar, validar y testear.

Se dividió el total de los datos en conjuntos de entrenamiento, validación y test.

Se debe tener en cuenta que al estar tratando series temporales y querer realizar predicciones a futuro, no se puede realizar una división de los datos de la misma forma que en un problema donde los datos no llevan ningún orden específico entre sí, como por ejemplo en un problema de clasificación de imágenes, donde se puede entrenar un modelo eligiendo datos al azar para cada uno de los conjuntos (lo cual es una buena práctica).

En un problema de series temporales uno quiere conocer la capacidad de extrapolar de un modelo, es decir de predecir datos del futuro. Por lo tanto para validar siempre se utilizan datos posteriores en el tiempo a los de entrenamiento.

Del total de datos, se ha utilizado el 70 % inicial para entrenar, el 20 % siguiente como conjunto de validación y el último 10 % como conjunto de test.

## 2.2. Implementación de modelos

Al momento de crear modelos predictivos es una buena práctica comenzar con modelos matemáticamente simples y de fácil comprensión, para obtener resultados iniciales que luego servirán como valores de referencia y que se intentarán sobrepasar con modelos más sofisticados. Con esta idea, se creó un modelo que para cada input dado por el valor de la acción y el volumen de la misma en  $n$  días consecutivos, se otorgue como predicción al último valor temporal del precio de la acción, es decir que da una predicción constante. Llamaremos a este último modelo base.

Debido a que los datos son secuencias ordenadas en el tiempo, se ha decidido utilizar una arquitectura de red neuronal que pueda considerar esta cualidad de los datos. Como consecuencia se ha utilizado redes neuronales recurrentes (RNN). En la figura 2 se puede ver la arquitectura utilizada, en el caso de una red recurrente de una capa.

Las RNN tienen consideración del orden de los datos utilizando conexiones entre neuronas de la misma capa, a diferencia de las redes feed-forward, que las conexiones de una neurona se realizan con neuronas ubicadas en otras capas.

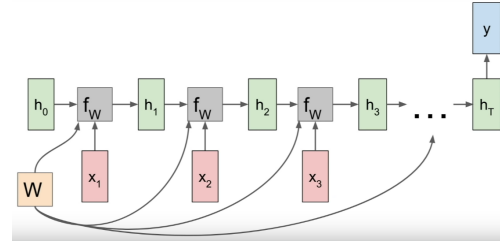


Figura 2: Arquitectura de una red neuronal recurrente de una capa con un único valor de salida.

Como se quiere predecir un único valor por cada input, se ha exigido que la red otorgue un único elemento unidimensional de salida utilizando con una capa fully connected al final de esta.

Para implementar los modelos se utilizó la librería Keras, donde para entrenar se utilizó la función de costo de error cuadrático medio (MSE), para realizar el descenso por el gradiente se ha utilizado el método Adam y se ha utilizado como funciones de activación a tangentes hiperbólicas.

Para tener una primer referencia del poder de extrapolación de las redes recurrentes, se utilizó un conjunto de datos dado por la función seno, que tiene claramente un comportamiento de simplicidad mayor que el comportamiento del precio de acciones.

Se aplicó el mismo preprocesamiento de datos que para los datos de Tesla y se entrenó una red recurrente sin realizar búsqueda de hiperparámetros, con 32 unidades (o neuronas) internas, 200 epochs y batch size de 200, sin dropout y una ventana de input de longitud 2.

Se comparó el poder predictivo de las redes con el modelo base, que extrapola tomando el último valor del input y otorgándolo como valor de salida. Los resultados obtenidos mostraron que la red recurrente ha obtenido resultados varios órdenes de magnitud mejores para los conjuntos de validación y test, lo cual se puede ver en la figura 3.

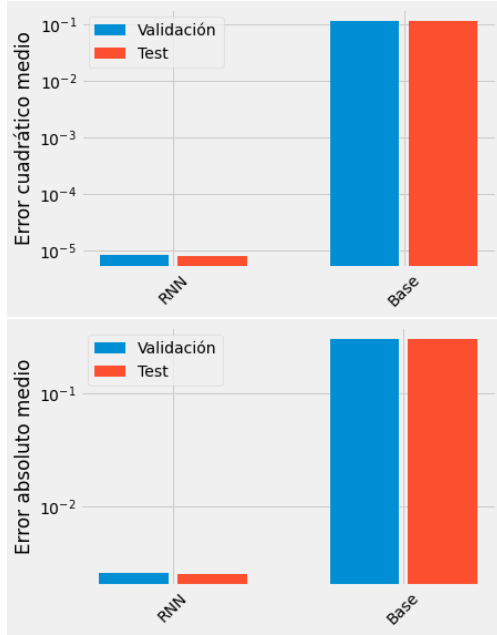


Figura 3: Resultados para la función seno

Una vez visto el poder predictivo de una RNN en un caso simple, se procedió al problema más complejo de predecir el precio de las acciones de Tesla.

Se ha utilizado conjuntos de datos con diferente cantidad de días que se utilizan como input en las redes neuronales (es decir, diferentes ventanas), con longitudes de 2, 60 y 360 días.

También se ha realizado una búsqueda de hiperparámetros para cada ventana, con la finalidad de encontrar un modelo con el mayor poder predictivo posible.

Para esto se confeccionó una grilla con los diferentes valores de hiperparámetros a probar. Luego se realizó una búsqueda sobre una gran cantidad de combinaciones de hiperparámetros, para tomar el modelo que mejor performance tenga sobre el conjunto de validación.

Una forma de realizar esto es a través de una búsqueda de hiperparámetros sobre todas las combinaciones posibles dentro de los rangos dados. Si la grilla es grande, el problema que puede ocasionar este enfoque es un alto costo computacional, ya que a través de fuerza bruta se evalúa la performance de todos los

modelos posibles. Una alternativa es realizar una búsqueda aleatoria dentro del conjunto de diferentes hiperparámetros para mitigar el problema del costo computacional (este método se llama Random Search). Cabe destacar que este método implica un tradeoff, ya que no utiliza todas las combinaciones posibles de hiperparámetros. Se requiere que la búsqueda sea lo suficientemente larga como para cubrir una buena parte de espacio de hiperparámetros en cuestión.

La grilla utilizada se ha conformado por:

- Capas recurrentes: 1, 2, 3 y 4.
- Neuronas internas de todas las capas recurrentes: 10, 50 y 100.
- Epochs: 50, 100, 200, 500 y 2000.
- Batch size: 50, 100 y 500.
- Dropout en la última capa: 0.0 y 0.3.
- Learning rate: 0.01, 0.001 y 0.0001.

Al realizar la búsqueda de hiperparámetros se ha entrenado una cantidad de modelos en el orden de miles.

Los mejores resultados se han obtenido con la ventana de 60 días e hiperparámetros batch size=50, dropout= 0.0, epochs=2000, capas= 1, learning rate = 0.0001 y neuronas internas = 50, utilizando un criterio de detención al no mostrar mejores resultados luego de 50 epochs. Se puede ver como ha aprendido la red a través de la figura 4.

Se puede ver que las RNN notablemente han aprendido, debido al decrecimiento de los errores sobre el conjunto en entrenamiento y validación al realizar el entrenamiento.

Luego, al momento de comparar los resultados con el modelo base, se dificulta que el aprendizaje sea suficiente como para que los modelos otorguen resultados mejores que las predicciones constantes, en cuanto MSE y MAE, que se puede ver en la figura 5.

Estos resultados están influenciados directamente por la complejidad del comportamiento de los datos.

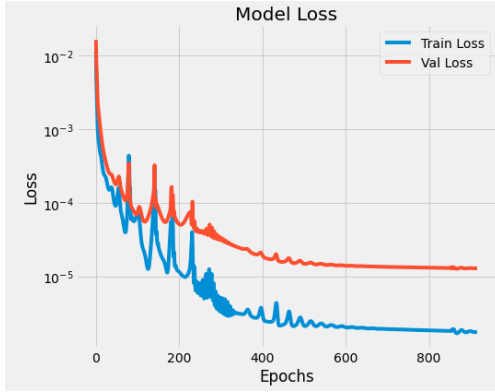


Figura 4: Error cuadrático medio del modelo con los mejores hiperparámetros sobre los conjuntos de entrenamiento y validación, graficado en escala logarítmica.

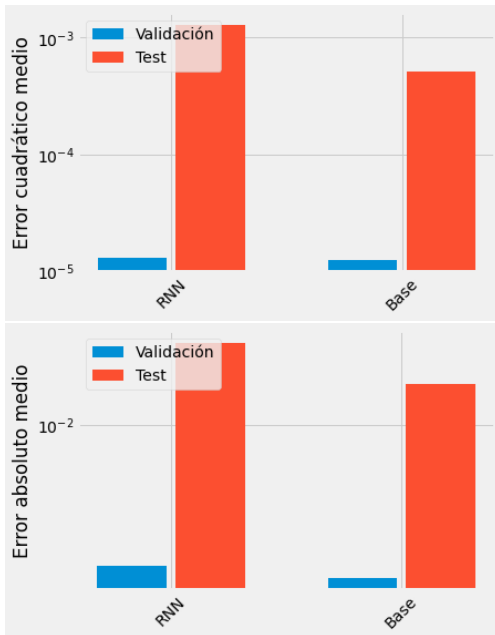


Figura 5: Comparación de los resultados de las predicciones de la mejor red recurrente con las del modelo base, en cuanto a MAE y MSE.

En particular para la acción de Tesla, notar que el comportamiento en los últimos años aumenta abruptamente, lo cual es difícil de modelar a través de datos de años anteriores.

Para analizar esta situación se debe tener en cuenta que las RNN asumen que los datos del

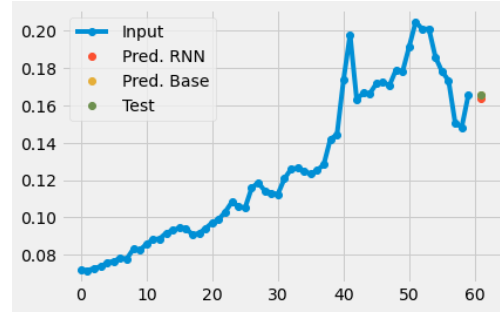


Figura 6: Muestra de una predicción de la acción de Tesla de la mejor red neuronal.

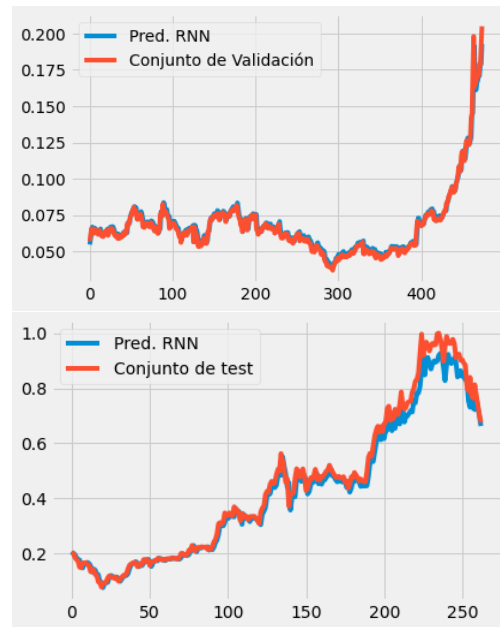


Figura 7: Comparación de las predicciones obtenidas sobre los conjuntos de validación y test con los datos del dataset.

pasado pueden influir en el presente a través de tendencias predecibles, lo cual parcialmente se cumple para datos de acciones, pero no siempre esto puede ser cierto, debido a la gran cantidad de factores que pueden influir en el precio, incluso de forma impredecible.

Notar que algunos de los factores de diversa índole que pueden influir en el precio de acciones pueden ser sucesos políticos (por ejemplo con la caída del índice Merval luego de las elecciones primarias, en agosto de 2019), even-

tos impredecibles como pandemias (en 2020, caída de del índice S&P 500 debido al Covid-19) o catástrofes naturales, por nombrar algunos, que no están siempre relacionados con el mercado. Incluso se conoce casos que hasta tweets de CEOs de empresas han logrado fluctuar precios (ha ocurrido con Elon Musk y Dogecoin, una criptomoneda).

Por otro lado, el libro "Deep Learning with Python"[1] enfatiza que no se debe utilizar técnicas usuales de series temporales para predecir el precio de acciones. También se pueden encontrar artículos [2], [3] que mencionan la gran dificultad de predecir acciones a través de redes recurrentes.

Sobre la complejidad de predecir los precios de acciones, se puede mencionar que en 1973, un economista de Princeton propuso que "un mono con los ojos tapados tirando dardos a las páginas financieras de un diario puede seleccionar un conjunto de acciones que tengan la misma performance que un conjunto seleccionado cuidadosamente por expertos"[4]. De hecho se han realizado simulaciones que han probado dicha afirmación. Estos resultados se pueden atribuir a que el precio de acciones tiene una componente que se puede considerar estocástica.

Estos son argumentos que muestran el porqué de la dificultad de ciertas redes neuronales al momento de predecir precios de acciones eficientemente de forma consistente.

### 3. Conclusión

Se ha realizado un estudio con redes recurrentes con la finalidad de predecir el precio de Tesla a 1 día. Las redes han logrado aprender considerablemente, pero no han logrado superar los resultados de una predicción constante en cuanto a MAE y MSE. Por lo tanto, los modelos utilizados no fueron lo suficientemente poderosos como para ser utilizados en operaciones de bolsa reales, donde se apunta a obtener ganancias económicas, y una mala performance del modelo puede generar pérdidas para el inversor.

En este trabajo ha sido de gran utilidad utilizar un modelo de complejidad baja para rea-

lizar predicciones iniciales que permitan servir de referencia para resultados de modelos más complejos, que como se ha visto, no necesariamente pueden otorgar mejores predicciones.

No se cierra la posibilidad de predecir precios de mejor forma con redes neuronales, aunque este es un problema que tiene cierto grado de complejidad.

Notar que en los dos dataset usados (el de la función seno y el de la acción de Tesla), el modelo base mantiene los errores sobre conjuntos de validación y test en el mismo orden de magnitud, pero esto no ocurre con las redes neuronales, que con los datos provenientes de la función seno aprenden muy bien y con los datos de acciones tiene errores del mismo orden que los obtenidos con predicciones constantes.

Posibles extensiones del trabajo son intentar predecir a más de un día en el futuro, utilizando más arquitecturas y de mayor complejidad, y nuevas ventanas de diferentes longitudes para conformar los datos de entrada de la red neuronal, aunque puede no esperarse resultados novedosos, debido a los argumentos presentados sobre la dificultad del problema.

También se puede considerar agregar features al dataset con información de noticias relacionadas, ya que se sabe que estos pueden ser factores influyentes en el precio de una acción. Incluso se puede probar con utilizar precios de otras acciones que puedan estar correlacionados con Tesla.

### 4. Referencias

- [1] Chollet, F. Deep learning with python - 2017 - Manning Publications.
- [2] No, LSTMs Can't Predict Stock Prices. <https://medium.com/analytics-vidhya/no-lstms-cant-predict-stock-prices-11f10dcb35d6>
- [3] Is it possible to predict stock prices with a neural network?. <https://towardsdatascience.com/is-it-possible-to-predict-stock-prices-with-a-neural-network-d750af3de50b>
- [4] Any monkey can beat the market. [www.forbes.com/sites/rickferri/2012/12/20/any-monkey-can-beat-the-market/?sh=3759e525630a](http://www.forbes.com/sites/rickferri/2012/12/20/any-monkey-can-beat-the-market/?sh=3759e525630a)