MODELO DE APRENDIZAJE REFORZADO APLICADO AL TRADING DE BITCOIN

Reinforced learning model applied to Bitcoin trading

SEBASTIAN OBANDO MORALES

Proyecto

Asesor

PhD Juan Rodrigo Jaramillo Posada

UNIVERSIDAD EAFIT

ESCUELA DE INGENIERÍAS

MAESTRÍA EN CIENCIAS DE LOS DATOS Y LA ANALÍTICA

MEDELLÍN

2022

Modelo de aprendizaje profundo reforzado aplicado al trading de bitcoin

Resumen

El mercado de valores se ve afectado por muchos tipos de factores, como el sentimiento del mercado, ir al alza (bulls) o a la baja (bears), el comportamiento de la economía o eventos políticos inesperados. Por tal razón, no es posible predecir su comportamiento, lo que significa que no es posible decidir cuándo entrar o cuando salir con certeza. Una aproximación como la del aprendizaje profundo reforzado, la cual puede emular la experiencia de un negociador (trader) quien no necesariamente predice precios, sino, momentos de entrada y salida del mercado, sería una opción viable. El presente trabajo buscó implementar una aproximación de aprendizaje profundo reforzado al trading de la bolsa (bitcoins, acciones y commodities), que haya demostrado resultados positivos en la literatura con retornos positivos sobre la inversión. El bot, resultado de este trabajo, obtuvo una rentabilidad del 5%. Estos resultados positivos abren la puerta a intentar nuevas aproximaciones que incluyan nuevas combinaciones en la forma de interpretar indicadores para encontrar estrategias ganadoras que aumenten la rentabilidad.

Descripción del Proyecto

1 Planteamiento del Problema

El proceso de adquirir y vender acciones en un intento de obtener ganancias del mercado reconocido comúnmente como *stock trading*, en donde, idealmente, las acciones deben recolectarse a un precio bajo y disponerse a un precio alto, para maximizar las ganancias, cuanto más alto sea el precio de venta que el precio de compra, mayor será la ganancia generada por la operación, siendo crítico determinar la oportunidad para entrar (comprar) o salir del mercado (vender) (Ee et al., 2020).

Es importante anotar que el mercado de valores se ve afectado por muchos tipos de factores, como el sentimiento del mercado, ir a la alza (bulls) o a la baja (bears), el comportamiento de la economía o eventos políticos inesperados (Ee et al., 2020). Por lo tanto, no es posible predecir su comportamiento, lo que significa que no es posible decidir cuándo entrar o cuando salir con certeza , tanto así, que el 90% de los traders pierden su capital, incluso apoyándose en sistemas profesionales (Corona-Bermudez et al.,2020), pero el 10% que lograr tomar ganancias, deben tener un conocimiento muy especializado para obtener estos resultados.

2 Justificación

Recientemente, los modelos de aprendizaje automático (ML), que forman parte de la inteligencia artificial (IA), se han utilizado para apoyar a los inversores y proporcionar un rendimiento de la inversión más significativo que las técnicas de análisis tradicionales por sí solas. Por lo tanto, la aplicación del aprendizaje de máquinas en el mercado de valores puede mejorar considerablemente el apoyo a las decisiones de los inversores y maximizar los beneficios, algunas veces incluso por encima del 20% (Brandao et al., 2020).

Para resolver el problema de optimizar las decisiones de trading, es decir, decidir cuándo entrar y cuándo salir, es posible, mediante su formulación, específicamente: dado k series de tiempo que describen el comportamiento histórico de K acciones y dado un presupuesto inicial, podemos determinar una política óptima de compra y venta para esos K instrumentos que maximice la ganancia (minimice las pérdidas), durante un horizonte de tiempo repartido en t franjas, esto en términos de una solución de aprendizaje profundo (Corona-Bermudez et al.,2020), se traduce en definir los espacios de acción y estado que naturalmente se ajustan a la manera como el 10% de los traders logran resultados positivos, analizan el entorno y toman decisiones.

En cuanto al Bitcoin como objeto de trading, este hace parte de los activos clasificados como criptomonedas o dinero digital basado en la tecnología de cadenas de bloques blockchain, el cual no depende de un gobierno o entidad pública emisora (Márquez, 2016) y según la revisión de la literatura, existen pocas investigaciones que se han realizado a nivel del uso de técnicas de aprendizaje profundo reforzado para hacer trading con Bitcoin (Bu & Cho, 2018) (Sattarov et al.,2020).

3 Objetivos

3.1 Objetivo general:

Implementar un modelo de aprendizaje profundo reforzado al trading de Bitcoin que haya demostrado resultados a nivel de retorno de inversión con balance positivo.

3.2 Objetivos específicos:

- Desarrollar la capacidad de comprender e interpretar indicadores técnicos cómo Moving Average (MA), Exponential Moving Average (EMA), Moving Average Convergence/Divergence (MACD), Bias, Volatility Volume Ratio (VR), On Balance Volume (OBV), para entender el estado del agente y por ende la razón de sus posibles acciones.
- Preparar la data a través del cálculo de los indicadores técnicos que ayudarán a determinar el estado del agente y las transformaciones necesarias para aprovecharla como input de forma eficiente en el proceso de modelado.
- Modelar la arquitectura del algoritmo DDPG (Deep Deterministic Policy Gradient) utilizando Python en conjunto con la librería Pytorch para materializar los agentes como productos de datos.
- Evaluar la rentabilidad del agente en función del retorno sobre la inversión, el cual mide el % de rendimiento del agente en función del ingreso o pérdida percibido por cada decisión.

4 Estado del arte y Marco teórico

Respecto a los conceptos y definiciones principales, el *aprendizaje por refuerzo* es una de las ramas del aprendizaje de máquinas, que busca representar computacionalmente el proceso de aprendizaje por ensayo y error de organismos biológicos (Sutton & Barto, 2018).

Dicha representación se realiza con el Proceso de Decisión Markoviano (MDP), el cual tiene 5 elementos (1) el agente (Agent), (2) el entorno (Environment), (3) la acción (Action), (4) el estado (State) y (5) la recompensa (Reward) (Figura 1).

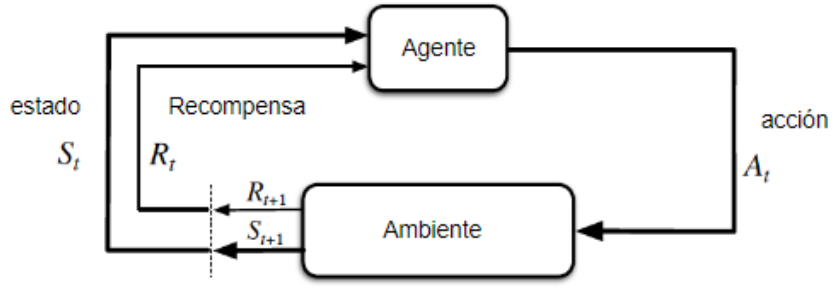

Figura 1. Interacción agente-ambiente en el aprendizaje reforzado (Sutton & Barto, 2018).

En dónde:

S: El estado en el cual se encuentra el agente.
A: La acción que toma el agente.
R: La recompensa percibida en función a la acción tomada y el estado en el cual se encuentra el agente.
t: El momento en el que se encuentra el agente.

Un ejemplo práctico de este proceso es el adiestramiento de un perro (agente), el cual mediante el refuerzo de una conducta esperada, como darle una galleta cada vez que da la pata, cuando se le pide que lo haga o corregirlo cuando no lo hace a criterio del adiestrador que haría las veces del entorno, llegará al punto de "aprender" la conducta sin necesidad de que exista siempre una recompensa, esta acción esperada por cada recompensa (darle una galleta o regañarlo) sería la política y cada punto en el que se encuentra el agente, en este caso el perro, para tomar una acción, es el estado.

Sin embargo, el MDP solo es un marco que define el proceso de aprendizaje reforzado, el problema a resolver radica en enseñarle al agente si las acciones que está tomando son las mejores o no. Este sexto elemento se logra encontrando la política óptima, para ello se emplea una aproximación llamada Q-learning (Hinton et al., 1995) (figura 2), la cual utilizando la función de valor o q function y siguiendo un algoritmo que itera entre cada uno de los estados y acciones posibles por estado, logra determinar el valor de cada decisión en función de la recompensa y comparar según ese valor cuál es la mejor acción. Al seguir la secuencia de decisiones que maximizan la recompensa, se logra encontrar la política óptima.

Algoritmo 14: Sarsamax (Aprendizaje-Q)

entrada: política π, entero positivo numero_episodios, pequeña fracción positiva α, $GLIE$ $\{\epsilon_i\}$
salida: valor de la función Q ($\approx q_\pi$ Si num_episodios es suficientemente grande)

Inicializar Q arbitrariamente (por ejemplo, $Q(s,a)$ = 0 para todos los $s \in S$ y $a \in A(s)$ y $Q(estado - final,.)$ = 0)

Para $i \leftarrow 1$ **hasta** numero_episodios **haga**

> $\epsilon \leftarrow \epsilon_i$
> Observe S_0
> $t \leftarrow 0$
> **repita**
>
> > Elija acción A_t usando la política derivada de Q(por ejemplo ϵ-ambicioso)
> > Tome acción A_t y observe R_{t+1}, S_{t+1}
> > $Q(S_t,A_t) \leftarrow Q(S_t,A_t) + \alpha(R_{t+1} + \gamma max_a Q(S_{t+1},a) - Q(S_t,A_t))$
> > $t \leftarrow t + 1$
>
> **hasta** S_t es $final$;

fin
devuela Q

Figura 2. Algoritmo de Q-learning (Sutton & Barto, 2018)

La desventaja del Q-learning al aplicarlo a problemas más reales, como ganarle al campeón de mundial de algún juego de mesa, radica en que el espacio de x estados por n acciones posibles puede ser demasiado

grande y puede resultar computacionalmente ineficiente calcular y comparar el valor de cada acción para cada estado con esta aproximación (Silver et al., 2017). Una alternativa para resolver esta limitante consiste en el empleo de redes neuronales, las cuales pueden encontrar óptimos locales del tipo mínimo o máximo para ecuaciones no lineales (Morales, 2020).

La repetición de la experiencia (Experience Replay) permite "etiquetar" experiencias previas, volviendo un problema del tipo supervisado el proceso de aprendizaje, seleccionando y almacenando acciones de forma aleatoria para disminuir correlaciones dañinas. De igual forma, los objetivos-Q ajustados, disminuyen adicionales potenciales correlaciones dañinas, ajustando los pesos de las redes neuronales para generar objetivos consistentes y así evitar adivinar un valor con otro valor adivinado (Morales, 2020).

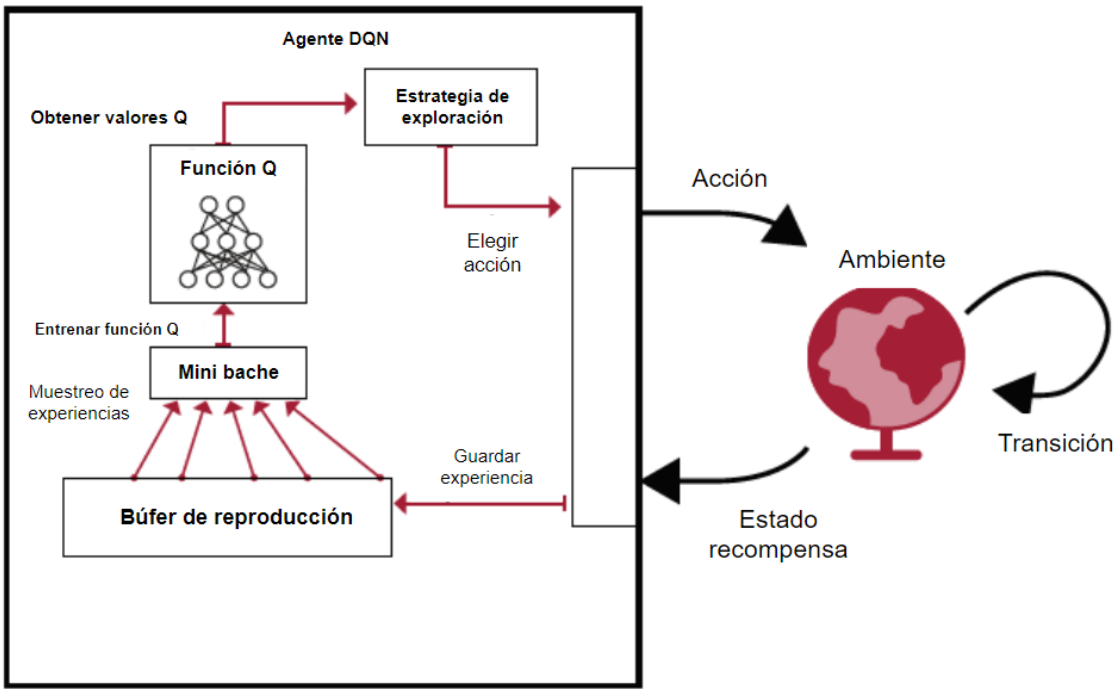

Figura 3. Arquitectura del proceso de aprendizaje reforzado profundo (Morales, 2020).

La repetición de la experiencia (Experience Replay) permite "etiquetar" experiencias previas, volviendo un problema del tipo supervisado el proceso de aprendizaje, seleccionando y almacenando acciones de forma aleatoria para disminuir dañinas correlaciones, de igual forma, los objetivos-Q ajustados, disminuyen adicionales potenciales correlaciones dañinas, ajustando los pesos de las redes neuronales para generar objetivos consistentes, para evitar adivinar un valor con otro valor adivinado (Morales, 2020).

Según la revisión de la literatura (Tabla 1), sobresalen los algoritmos del tipo optimización de valor como el DQN (Figura 4) y Actor - Crítico como el DDPG (Figura 5), con retornos sobre la inversión para el primer caso de 8258% (Ee et al., 2020), 2087% (Bu & Cho, 2018), 857% (Leem & Kim, 2020), 399% (Chakole & Kurhekar, 2020) y para el segundo de 311% (Conegundes & Pereira, 2020), de los cuales 2 operaron en

índices de Estados Unidos (S&P 500 y NASDAQ), 1 en Brazil (Ibovespa), 1 en criptomonedas (BITCOIN) y 1 sobre la acción de Google (GOOGL). También, de esos 5, 4 presentan variaciones en el algoritmo Deep Q Network (DQN) y 1 utilizó el algoritmo Deep Deterministic Policy Gradient (DDPG).

Artículo	Citaciones	Opción de Inversión	Periodo de prueba	Mejor Técnica	Mejor Resultado
(Ee et al., 2020)	0	EE.UU - GOOGL	Ene 2015 - Dic 2016	RDQN	8258%
(Bu & Cho, 2018)	2	BITCOIN	Dic 2017 - Ene 2018	DBM - DQN	2087%
(Leem & Kim, 2020)	1	EE.UU - S&P500	2008 - 2018	DQN	857%
(Chakole & Kurhekar, 2020)	1	EE.UU - NASDAQ	Ene 2006 - Dic 2018	DQN	399%
(Conegundes & Pereira, 2020)	0	BRAZIL - Ibovespa	Ene 2015 - Dic 2019	DDPG	311%
(Li et al., 2019)	14	EE.UU - APPL	Ene 2008 - Ene 2018	SDAEs-LSTM A3C	85%
(Wu et al., 2020)	8	EE.UU - APPL	Ene 2016 - Dic 2018	GDPG	82%
(Sattarov et al.,2020)	1	LITECOIN	Mar 2019 - Abr 2019	DQN	74%
(Yuan et al., 2020)	0	CHINA - NDSD	2016	PPO	66%
(Liu et al., 2020)	1	EE.UU - OHLC	May 2018 - May 2019	iRDPG	38%
(Théate & Ernst, 2021)	0	EE.UU - APPL	Ene 2018 - Dic 2019	TDQN	33%
(Chen & Gao, 2019)	0	EE.UU - S&P500	Ene 2005 - Dic 2018	DRQN	23%
(Lee et al., 2020)	0	Russel 300 index	2006 - 2018	MAPS	23%
(Lei et al., 2020)	9	EE.UU - S&P500	2007 - 2018	TFJ-DRL	22%

Tabla 1. Revisión de la literatura.

Para el caso del Deep Q Network (DQN) (Mnih et al., 2015), estos traducen la denominada Q-table y resuelven la función de valor a través de una red neuronal (Figura 4), recibiendo como input la variables que describen el estado (*state*) del agente en el momento *t* y generando como output los valores de cada posible acción (*a*) en dicho estado de la forma $Q(S_t,a_n)$.

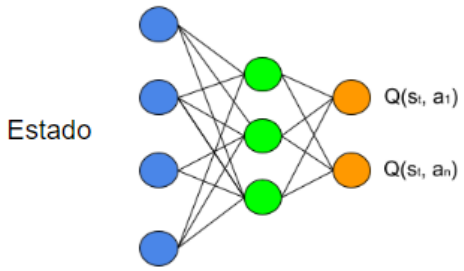

Figura 4. Arquitectura del DQN (Morales, 2020).

Por otro lado, el Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2016), utiliza dos redes neuronales, (1) el actor y (2) el crítico (Figura 5). Para este caso, se utilizan tanto la aproximación del DQN haciendo las veces de crítico, como el denominado Proximal Policy Optimization (PPO) (Schulman et al.,

2017) haciendo las veces de actor, de manera que mientras el actor va tratando de encontrar una política, el crítico va evaluando que tan buena o mala es la calidad de las decisiones de dicha política.

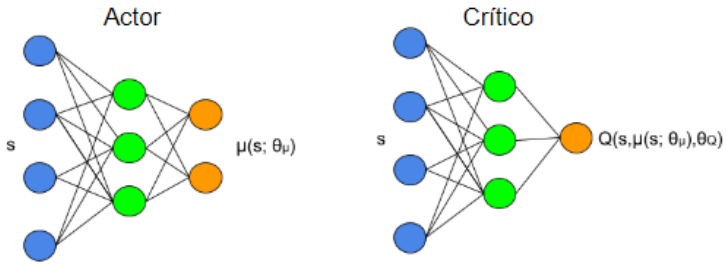

Figura 5. Arquitectura del DDPG (Morales, 2020).

5 Metodología

En este trabajo se hace una adaptación de la CRISP-DM (Shearer, 2000) de la siguiente manera:

Datos: date, price open, high, close, adj close y volumen de 2018 a 2022 de Bitcoin, Apple (APPL), El índice de las 500 compañías más grandes en la bolsa de Nueva York (SPY) y el Oro y la Plata (^AUX) provenientes de yahoo finance (finance.yahoo.com).

1. **Entendimiento del negocio**:
   El propósito de esta etapa fue adquirir la capacidad de interpretación los indicadores técnicos (señales) a utilizar para la determinación del estado (S_t) del agente, tales como Moving Average (MA), Exponential Moving Average (EMA), Moving Average Convergence/Divergence (MACD), Bias, Volatility Volume Ratio (VR), On Balance Volume (OBV).

2. **Entendimiento de los datos:**
   Se realizó un análisis exploratorio de los datos con y sin indicadores, para entender la estructura, relaciones y presentación de los datos, con el fin de identificar oportunidades de limpieza y posibles alertas (insights) tempranas que ayudaron a explicar comportamientos durante el modelado y evaluación.

3. **Preparación de los datos:**
   Se calcularon los indicadores técnicos para cada stock y se realizaron las transformaciones que fueron necesarias para limpiar y presentar los datos como la etapa anterior lo determinó.

4. **Modelamiento:**
   Se implementó una arquitectura del tipo actor-crítico con DDPG utilizando el lenguaje de programación Python con la librería Pytorch, corriendo el modelo con un máximo de 106 episodios durante 24 horas.

5. **Evaluación:**

Se evaluó el rendimiento del bot utilizando los modelos entrenados para el actor y el crítico, sobre el mismo set de datos de entrenamiento, teniendo en cuenta que el modelo no es un modelo que predice o clasifica a partir de unas variables de entrada, sino que optimiza en busca de máximos a partir de unas reglas de castigo y recompensa.

6 Resultados

Para el desarrollo del proyecto se empleó una suscripción a Colab Pro +, con la configuración:

- Hardware accelerator: GPU
- Runtime shape: High-RAM
- Background execution

Y se creó una serie de notebooks bajo una arquitectura dividida en 4 “Fases” (1) Análisis, (2) Preprocesamiento (3) Repositorio y (4) Modelamiento (Figura 6) como se muestra a continuación:

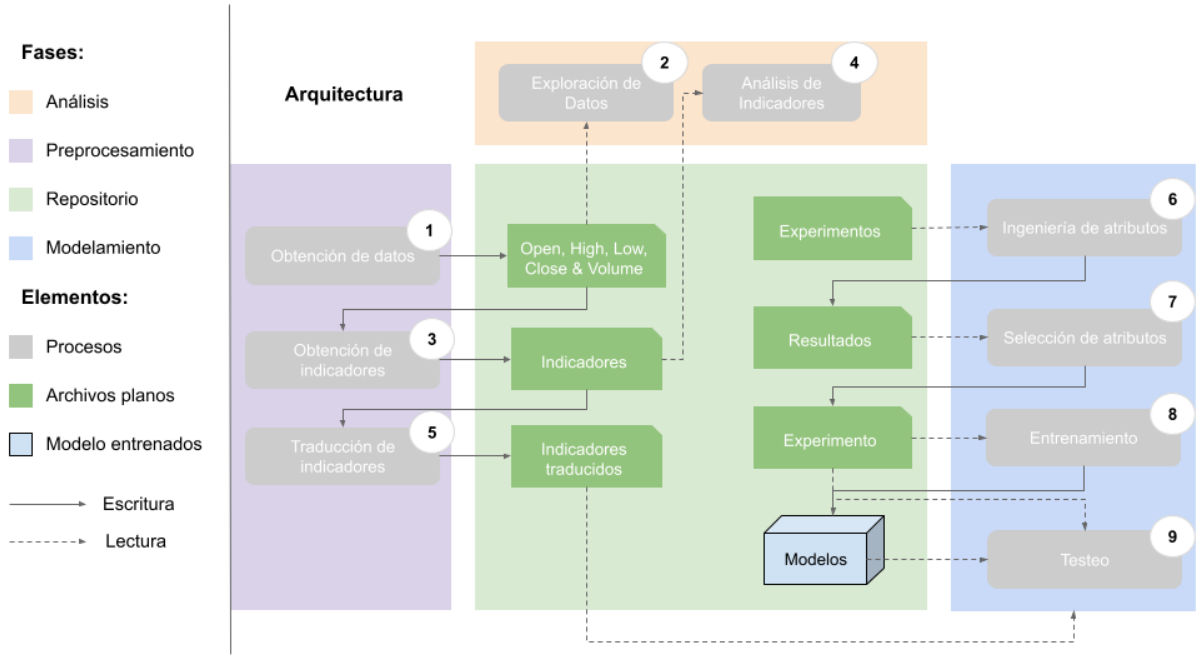

Figura 6. Arquitectura de la solución

A su vez los componentes se clasificaron en una serie de “Elementos” (1) Procesos, (2) Archivos Planos y (3) Modelos entrenados.

De la fase de análisis exploratorio se evidenció que el precio del Bitcoin creció unas 6 veces en el 2021 frente a lo que venía del 2019 y estuvo más o menos estable con un valor oscilante entre 10 mil y 20 mil dólares para el periodo 2018 hasta principios del 2021.

Respecto al volumen, las transacciones estuvieron en el orden de 10 a la 11 operaciones por día en los 1518 días del dataset, habiendo un valor atípico de 3.5 x 10 a la 12 transacciones frente a un límite de 1 x

10 a la 11 transacciones, lo que pudo contribuir a que creciera de forma significativa el valor del precio durante el 2021 (Figura 8).

Figura 8. Precio vs Volumen 2018 a 2022

Y en el análisis de indicadores, para el caso de las medias móviles se pudo deducir una estrategia de entrada cuando las medias móviles de 50 días están por encima de las medias móviles de 200 días y de salida cuando pasa lo contrario, por ejemplo hubo una oportunidad clara de entrada en el segundo trimestre del 2020 cuando el precio y las medias móviles de 50 días sobrepasan las de 200 días, con una señal de salida casi que al año cuando las medias móviles de 50 días alcanzan su máximo del primer semestre y oscilan por debajo y por encima de las medidas móviles de 200 días en el 2021 (Figura 9).

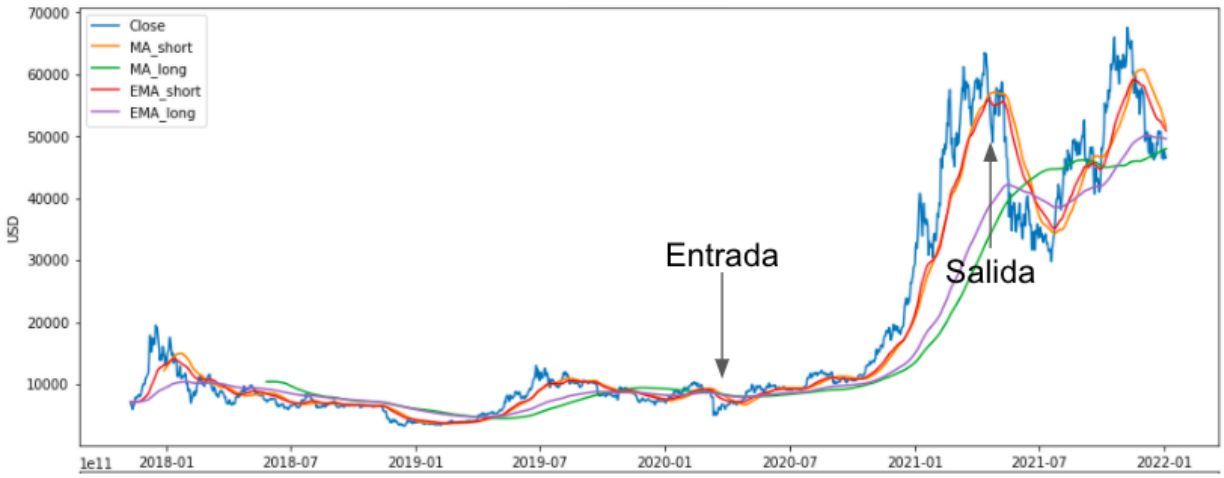

Figura 9. Medias móviles vs el precio y el volumen 2018 a 2022

Sin embargo, para el caso del resto de los indicadores no es claro establecer que variaciones pueden determinar en qué momento entrar o salir y debido a esta potencial complejidad para deducir las

relaciones entre los indicadores y el precio o el volumen, se implementó una aproximación para interpretar los indicadores en función del precio y el volumen en tres categorías -1, 0, 1 siendo -1 la categoría del indicador que está por debajo, 0 igual y 1 superior, de esta manera lograr una convergencia fue más probable, dado que los estados al no ser tan cambiantes en el tiempo le pueden permitir al bot, comprobar que combinación de categorías por indicador pueden representar una señal de entrada o salida.

Cabe resaltar que el único indicador que tiene una particularidad, es decir, no necesariamente se compara directamente con el precio o el volumen es el MACD. Este indicador calcula una serie de promedios móviles ponderados a razón de 9, 12 y 26 días, y mide las diferencias entre el de 12 y 26 y compara la diferencia respecto al de 9.

Una vez definida la interpretación de los indicadores en términos de estados concretos que pueden repetirse en el tiempo en lugar de ser distintos, se utilizó una batería de experimentos de 32 combinaciones en términos de la arquitectura de las redes neuronales, el número de nodos por capa y los periodos de actualización y aprendizaje a razón de 1 episodio para evaluar la viabilidad computacional con los recursos disponibles de cada combinación de variables por experimento (Tabla 2).

Variable	Nivel 1	Nivel 2	Nivel 3	Nivel 4
Tipo de cantidad por capa	1	2		
Tipo de capas	Constant	Decreasing		
Nodos	128	512	600	1024
Periodo de aprendizaje	5	20		
Factor de actualización	1	5		
Mini bache	512	1024	2048	

Tabla 2. Niveles y valores por variable

En donde:

Tipo de cantidad por capa: Son los dos tipos de configuraciones de redes usados, para el Nivel 1 como una combinación de 5 capas en la red del actor y 6 en la del crítico y el Nivel 2, 3 capas en el actor y 4 en el crítico.
Tipo de capas: el tipo de capas si se mantenían con la misma cantidad de nodos (Constant) o si variaban (Decreasing).
Nodos: La escala de nodos a usar en las redes.
Periodo de aprendizaje: El periodo de aprendizaje en días.
Factor de actualización: El factor de aprendizaje en días.
Mini bache: El tamaño del mini bache.
Para posteriormente, seleccionar el experimento con el mejor rendimiento en términos del “scoring”, promedio y retorno sobre la inversión (resaltado en verde en la Figura 10).

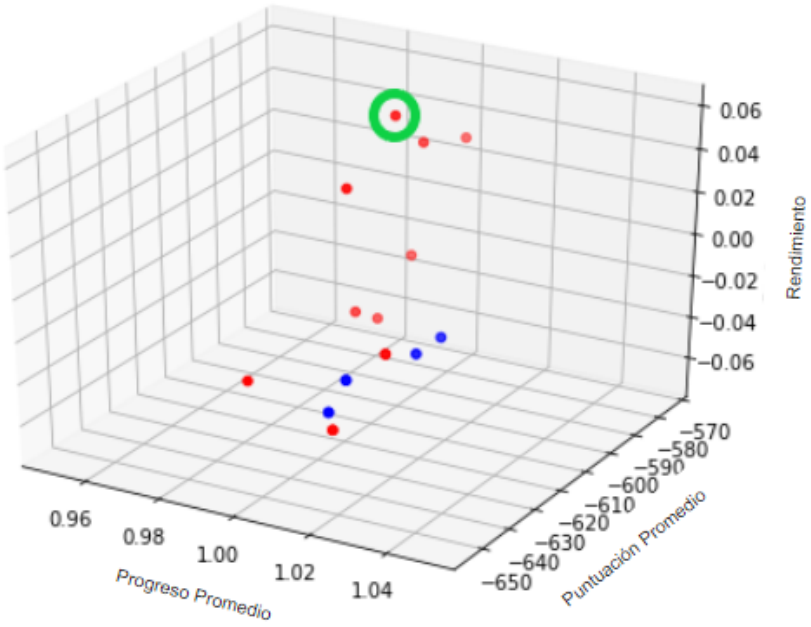

Figura 10. Selección del mejor modelo

La combinación de parámetros seleccionada fue:

Tipo de cantidad por capa: 1
Tipo de capas: Constant
Nodos: 128
Periodo de aprendizaje:5
Factor de actualización: 5
Mini bache: 512

Con la arquitectura del modelo definido se estableció una corrida de 1000 episodios a razón de los dos últimos años de datos (Figura 11), pero la capacidad máxima de Colab Pro + solo alcanzó los 106 episodios que al momento de evaluar el modelo logró un retorno de la inversión del 5% sobre 10 veces el valor más alto del precio en ese periodo de tiempo.

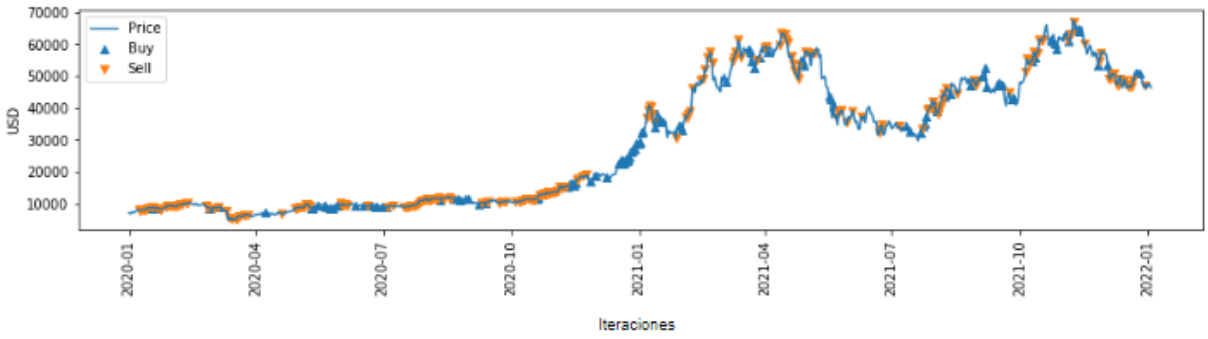

Figura 11. Comportamiento del modelo evaluado

De igual manera se implementó el modelo a otra serie de stocks como lo son las acciones de Apple (APPL), el Standard & Pool 500 (SPY) y el oro y la plata (^XAU), pero para todos los casos con rendimientos negativos (Tabla 2).

Stock	Inversión Inicial (USD)	Ganancia / Pérdida (USD)	Retorno sobre la inversión (%)
BTC-USD	687,896.25	31,386.75	0.054
APPL	1828.80	122.28	-0.066
SPY	4,799.80	375.96	-0.078
^XAU	1,670.90	46.29	-0.027

Tabla 2. Rendimiento del bot en distintos stocks

Los stocks utilizados distintos al Bitcoin, solo se tuvieron en cuenta como una manera de medir la capacidad de generalización del modelo, por lo que se elige un índice como el SPY, una acción muy popular cómo APPL y commodities como el oro y la plata.

7 Conclusiones

Se puede decir que el tipo de inversión del bot es de bajo riesgo o muy conservadora, dado que los niveles de pérdida estaban por debajo del 10% para períodos medidos en años y puntualmente para el caso del Bitcoin logró un 5 % de retorno, siendo la interpretación de los indicadores a utilizar una pieza fundamental del resultado.

No obstante, no se logra una estabilidad para llevar el bot a producción dado el nivel de iteraciones limitado que los recursos disponibles ofrecen.

Finalmente se abre la posibilidad a buscar otras estrategias de entrenamiento, como reducir los periodos de tiempo de entrenamiento, generar series de tiempo sintéticas que le permitan deducir estrategias de entrada y salida con menos recursos de cómputo y mayor estabilidad y experimentar con la función de recompensa para volver más agresivo el bot, lo cua podría contribuir en resultados más favorables.

8 Productos esperados

El código del proyecto se encuentra en: https://github.com/seobando/TradingBot

9 Plan de gestión de datos

Los datos originales no serán alterados de alguna forma, las transformaciones o valores adicionales que se generen a partir de la implementación del modelo desarrollado por el presente trabajo serán

almacenados como set de datos distintos a estos originales, en archivos de tipo csv en el equipo de cómputo del autor del presente trabajo.

10 Aspectos éticos

Los datos serán utilizados para entrenar un modelo de aprendizaje profundo reforzado en la toma de decisiones de compra y venta de Bitcoins, en el equipo de cómputo del autor del presente trabajo, quien será el beneficiario de los resultados que este ejercicio generen.

11 Referencias bibliográficas

Brandao, I. V., Da Costa, J. P. C. L., Praciano, B. J. G., De Sousa, R. T., & De Mendonca, F. L. L. (2020). Decision support framework for the stock market using deep reinforcement learning. In 2020 Workshop on Communication Networks and Power Systems, WCNPS 2020. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/WCNPS50723.2020.9263712

Bu, S. J., & Cho, S. B. (2018). Learning Optimal Q-Function Using Deep Boltzmann Machine for Reliable Trading of Cryptocurrency. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 11314 LNCS, pp. 468–480). Springer Verlag. https://doi.org/10.1007/978-3-030-03493-1_49

Chakole, J., & Kurhekar, M. (2020). Trend following deep Q-Learning strategy for stock trading. Expert Systems, 37(4). https://doi.org/10.1111/exsy.12514

Chen, L., & Gao, Q. (2019). Application of deep reinforcement learning on automated stock trading. In Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS (Vol. 2019-October, pp. 29–33). IEEE Computer Society. https://doi.org/10.1109/ICSESS47205.2019.9040728

Conegundes, L., & Pereira, A. C. M. H. (2020). Beating the Stock Market with a Deep Reinforcement Learning Day Trading System. In Proceedings of the International Joint Conference on Neural Networks. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/IJCNN48605.2020.9206938

Corona-Bermudez, U., Menchaca-Mendez, R., & Menchaca-Mendez, R. (2020). On the computation of optimized trading policies using deep reinforcement learning. In Communications in Computer and Information Science (Vol. 1280, pp. 83–96). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/978-3-030-62554-2_7

Ee, Y. K., Sharef, N. M., Yaakob, R., & Kasmiran, K. A. (2020). LSTM Based Recurrent Enhancement of DQN for Stock Trading. In 2020 IEEE Conference on Big Data and Analytics, ICBDA 2020 (pp. 38–44). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICBDA50157.2020.9289832

Hinton, G. E., Dayan, P., Frey, B. J., & Neal, R. M. (1995). The “wake-sleep” algorithm for unsupervised neural networks. Science, 268(5214), 1158–1161. https://doi.org/10.1126/science.7761831

Lee, J., Kim, R., Yi, S. W., & Kang, J. (2020). MAPS: Multi-agent reinforcement learning-based portfolio management system. In IJCAI International Joint Conference on Artificial Intelligence (Vol. 2021-January, pp. 4520–4526). International Joint Conferences on Artificial Intelligence. https://doi.org/10.24963/ijcai.2020/623

Leem, J. B., & Kim, H. Y. (2020). Action-specialized expert ensemble trading system with extended discrete action space using deep reinforcement learning. PLoS ONE, 15(7 July). https://doi.org/10.1371/journal.pone.0236178

Lei, K., Zhang, B., Li, Y., Yang, M., & Shen, Y. (2020). Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading. Expert Systems with Applications, 140. https://doi.org/10.1016/j.eswa.2019.112872

Li, Y., Ni, P., & Chang, V. (2019). An Empirical Research on the Investment Strategy of Stock Market based on Deep Reinforcement Learning model. In COMPLEXIS 2019 - Proceedings of the 4th International Conference on Complexity, Future Information Systems and Risk (pp. 52–58). SciTePress. https://doi.org/10.5220/0007722000520058

Li, Y., Zheng, W., & Zheng, Z. (2019). Deep Robust Reinforcement Learning for Practical Algorithmic Trading. IEEE Access, 7, 108014–108021. https://doi.org/10.1109/ACCESS.2019.2932789

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., … Wierstra, D. (2016). Continuous control with deep reinforcement learning. In 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings. International Conference on Learning Representations, ICLR.

Liu, Y., Liu, Q., Zhao, H., Pan, Z., & Liu, C. (2020). Adaptive quantitative trading: An imitative deep reinforcement learning approach. In AAAI 2020 - 34th AAAI Conference on Artificial Intelligence (pp. 2128–2135). AAAI press. https://doi.org/10.1609/aaai.v34i02.5587

Marquez, S. (2016). Bitcoin: Guía completa de la moneda del futuro.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., … Hassabis, D. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529–533. https://doi.org/10.1038/nature14236

Morales, Miguel. (2020). Grokking Deep Reinforcement Learning. Manning Publications; 1er edición

Sattarov, O., Muminov, A., Lee, C.W., Kang, H.K., Oh, R., Ahn, J., Oh, H.J., & Jeon, H. (2020). Recommending Cryptocurrency Trading Points with Deep Reinforcement Learning Approach. Applied Sciences, 10, 1506.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal Policy Optimization Algorithms. Retrieved from http://arxiv.org/abs/1707.06347

Shearer, C. (2000). The CRISP-DM Model: The New Blueprint for Data Mining. Journal of Data Warehousing 5(4).

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T. & Hassabis, D. (2017). Mastering the game of Go without human knowledge. Nature, 550, 354.

Théate, T., & Ernst, D. (2021). An application of deep reinforcement learning to algorithmic trading. Expert Systems with Applications, 173. https://doi.org/10.1016/j.eswa.2021.114632

Thrun, S., & Schwartz, A. (1993). Issues in Using Function Approximation for Reinforcement Learning. Proceedings of the 4th Connectionist Models Summer School Hillsdale, NJ. Lawrence Erlbaum, 1–9.

Wu, X., Chen, H., Wang, J., Troiano, L., Loia, V., & Fujita, H. (2020). Adaptive stock trading strategies with deep reinforcement learning methods. Information Sciences, 538, 142–158. https://doi.org/10.1016/j.ins.2020.05.066

Yuan, Y., Wen, W., & Yang, J. (2020). Using data augmentation based reinforcement learning for daily stock trading. Electronics (Switzerland), 9(9), 1–13. https://doi.org/10.3390/electronics909