



APRENDIZAJE REFORZADO PARA ESTRATEGIAS DE TRADING

Por

Darío Castro González

Fernando De Santos Franco

Sergio Daniel Dueñas Godínez

André Yahir González Cuevas

Flavio Maximiliano Herrada Avalos

Ernesto Morales Mozka

Ingeniería Financiera

Instituto Tecnológico y de Estudios Superiores de Occidente

2025

Revisado por

Luis Felipe Gómez Estrada

1. Introducción

El Aprendizaje por Refuerzo (Reinforcement Learning, RL) es una técnica efectiva para la toma de decisiones secuenciales en entornos complejos como los mercados financieros. A diferencia de otros enfoques de Machine Learning, el RL permite a un agente aprender a través de la interacción con el entorno, ajustando sus decisiones según las recompensas obtenidas.

Este proyecto utiliza el algoritmo Q-Learning para desarrollar un agente capaz de tomar decisiones de trading basadas en indicadores técnicos como SMA, RSI y MACD. Los estados se construyen a partir de estos indicadores, mientras que las acciones corresponden a comprar, vender o mantener activos. La recompensa se define según las ganancias netas tras cada operación, considerando comisiones por transacción.

El agente utiliza una política epsilon-greedy para equilibrar la exploración y la explotación durante el entrenamiento, asegurando que se exploren nuevas estrategias en fases iniciales y se prioricen las más efectivas a medida que aprende.

Esta entrega describe los componentes clave del sistema, los resultados preliminares y los próximos pasos para mejorar el modelo, como la implementación de Deep Q-Learning para manejar mejor los espacios de estado continuos.

2. Estructura del Código

El proyecto se organiza en archivos principales:

1. **data_base.py**: Descarga y procesamiento de los datos históricos financieros.
2. **environment.py**: Definición del entorno de trading, usando OpenAI gym
3. **qlearning.py**: Implementación del deep Q-Learning con stable-baselines3.
4. **training.py**: Entrenamiento y evaluación del agente.

3. Descripción de los Módulos

3.1 data_base.py

- **Función download_market_data**: Descarga datos históricos desde Yahoo Finance mediante la API de yfinance. La precisión y disponibilidad de los datos son factores críticos para garantizar la efectividad del agente.
- **Función calculate_indicators**: Utiliza la librería ta para calcular indicadores técnicos como las medias móviles simples (SMA), el índice de fuerza relativa (RSI) y el MACD, que son fundamentales para la toma de decisiones basada en tendencias y momentos del mercado.

3.2 environment.py

- **TradingGymEnv:** Un entorno personalizado de OpenAI Gym que simula operaciones de trading. **Características clave:**
 - Representación del estado que combina precio, indicadores técnicos y estado del portafolio.
 - Espacio de acción discreto (Mantener, Comprar, Vender).
 - Mecanismo de recompensa basado en cambios en el valor del portafolio y costos de transacción.

3.3 deepQLearning.py

- **Implementa DQN de la librería Stable Baselines3.**
- **Estrategia** avanzada de exploración:
 - Exploración epsilon-greedy.
 - Reducción gradual de la tasa de exploración.
 - Replay buffer para un aprendizaje estable.
- Uso de redes neuronales para aproximar los valores de estado-acción.

3.4 training.py

- Coordina el **entrenamiento, evaluación y visualización** del modelo.
- Genera **métricas de rendimiento** y gráficos.
- Guarda el modelo entrenado y los datos de desempeño.

4. Enfoque técnico.

- **Algoritmo de Aprendizaje**
Algoritmo: Deep Q-Learning (DQN)
 - Política: Basada en redes neuronales con exploración epsilon-greedy.
 - Espacio de estado: Vector de 7 dimensiones que incluye:
 - Precio actual.
 - Indicadores técnicos (SMA_50, SMA_200, RSI_14, MACD).
 - Número de acciones en posesión.
 - Saldo actual.
- **Hiperparámetros Clave**
 - Tasa de aprendizaje: 1e-3
 - Tasa de exploración: 1.0 → 0.05
 - Factor de descuento (gamma): 0.99
 - Tamaño del replay buffer: 10,000
 - Batch size: 64

5. Conclusión

Esta implementación de Deep Q-Learning proporciona un marco sólido para el desarrollo de estrategias de trading algorítmico. Mediante redes neuronales y aprendizaje por refuerzo, el agente puede adaptarse a dinámicas complejas del mercado y aprender estrategias de trading no lineales.

El diseño modular facilita la experimentación y mejora continua, sirviendo como una base robusta para futuras investigaciones en trading cuantitativo avanzado.

8. Referencias

- Fan, J., Wang, Z., Xie, Y., & Yang, Z. (2020). A Theoretical Analysis of Deep Q-Learning. Retrieved from <https://proceedings.mlr.press/v120/yang20a>
- Hyungjun, P., Min-Kyu, S., Dong-Gu, C. (2020). An intelligent financial portfolio trading strategy using deep Q-learning. Retrieved from <https://www.sciencedirect.com/science/article/abs/pii/S0957417420303973>
- Yahoo Finance (2025). yfinance 0.2.54. Retrieved from <https://pypi.org/project/yfinance/>
- Darío López Padial (n.d.). Technical Analysis Library in Python. Retrieved from <https://technical-analysis-library-in-python.readthedocs.io/en/latest/>
- Morales, S. O. (s/f). MODELO DE APRENDIZAJE REFORZADO APLICADO AL TRADING DE BITCOIN. Edu.co. Recuperado el 11 de febrero de 2025, de <https://repository.eafit.edu.co/server/api/core/bitstreams/45d87400-5d0f-42e2-bed6611b9b5f915f/content>