Algoritmo de Trading Usando Aprendizado por Reforço para Operações em Mini Índice

Seminário ABEX - UNOCHAPECÓ

Álvaro G. Pagliari
Miliano F. de Oliveira Jr.

Introdução

- Algoritmo de Trading
  - O Trading algorítimico, refere-se ao uso de programas de computador e algoritmos complexos para negociar ativos financeiros de forma automática.
  - Surgiu nas décadas de 70 e 80 com a disseminação de computadores pessoais e redes eletrônicas de negociação.
  - Os algoritmos são programados para tomar decisões de compra e venda baseadas em uma série de fatores, incluindo sinais de mercado, padrões de preços históricos, notícias econômicas, entre outros.
- Aprendizagem por Reforço
  - É uma subárea da inteligência artificial onde os algoritmos aprendem a tomar decisões ótimas através de tentiva e erro, sendo recompensados por ações corretas. É particularmente útil em ambientes incertos e dinâmicos, como o mercado financeiro.
  - O aprendizado por reforço pode otimizar estratégias de trading, ajustando-se automaticamente às mudanças do mercado.

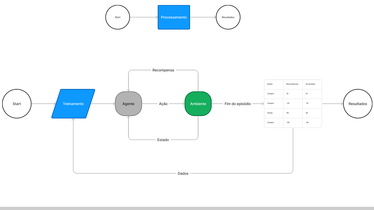
Algoritmos de Trading

- Os algoritmos de trading são capazes de comprar e vender a partir de dados de entrada pré definidos sem intervenção humana.
- Vantagens
  - Velocidade e eficiência
  - Disciplina e consistência
  - Capacidade de backtesting
  - Diversificação
- Desafios
  - Complexidade na criação de estratégias
  - Risco de sobre-otimização (overfitting)
  - Latência
  - Mudanças de mercado
  - Aspectos regulatórios

Aprendizado por reforço

- Aprendizado por Reforço (Reinforcement Learning - RL) é um tipo de aprendizado de máquina onde um **agente** aprende a tomar decisões, mapeando situações (**estados**) a **ações**, com o objetivo de maximizar uma **recompensa** cumulativa.
- O aprendizado ocorre através de trial-and-error, onde o agente experimenta diferentes estratégias. É diferente de outros aprendizados de máquina porque o agente tem interação direta com um **ambiente**, sem ser explicitamente programado para realizar tarefas específicas.
- Componentes Chave
  - **Agente**: entidade que toma decisões
  - **Ambiente**: o contexto ou o mundo no qual o agente opera e toma decisões
  - **Recompensas**: sinais numéricos que o agente recebe do ambiente como resposta às suas ações
  - **Estados**: representações da situação atual em que o agente se encontra.
  - **Ações**: todas as possíveis movimentações ou decisões que o agente pode tomar no ambiente.

Aprendizado por reforço



Aplicação de Aprendizado por Reforço em Trading

- Por que o Aprendizado por Reforço é adequado ao Trading Algorítmico?
  - Adaptação dinâmica: o RL permite que os modelos se adaptem continuamente a novas condições de mercado, aprendendo com a experiência anterior.
  - Decisões baseadas em recompensa: alinha-se perfeitamente com o objetivo do trading, que é maximizar o retorno sobre o investimento.
  - Exploração vs Exploração: o RL equilibra naturalmente a exploração de novas estratégias com a exploração de abordagens já conhecidas e lucrativas.
  - Processamento de sinais complexos: pode processar e aprender uma vasta quantidade de dados de mercado, incluindo séries temporais de preços, volumes, indicadores técnicos e notícias.
- Exemplos de estratégias
  - Otimização de Execução de Ordem
  - Seleção de portfólio e rebalanceamento
  - Arbitragem estatística
  - Trading de Alta Freqüência (HFT)

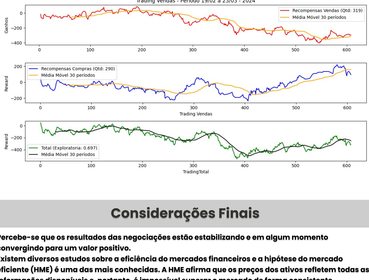
Modelos e Algoritmos de RL

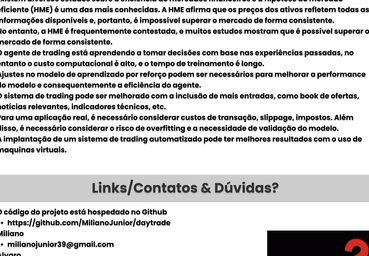
- Q-Learning
  - Resumo: É um algoritmo de aprendizagem por reforço baseado em valores que visa aprender a qualidade de ações, indicando o quão útil é uma ação para obter alguma recompensa futura.
  - Aplicação: Determina a melhor ação (compra, venda, manutenção) em um determinado estado do mercado
- Deep Q Network (DQN)
  - Resumo: Combina Q-Learning com redes neurais profundas. DQN pode lidar com espaços de estado e ação de alta dimensão, permitindo que o algoritmo aprenda políticas ótimas em ambientes complexos.
  - Aplicação: Aprende políticas de trading em um ambiente altamente estocástico e multidimensional, como o mercado financeiro
- Policy Gradients
  - Resumo: ajusta diretamente os parâmetros da política de ação do agente, em vez de aprender um valor de ação
  - Aplicação: Usado quando a política de ação é complexa e não pode ser facilmente mapeada para estados específicos.

Implementação

- Os dados de negociação foram obtidos através da API do MetaTrader
- A implementação utilizou Python como linguagem de programação e o framework Tensor Flow para treinamento com GPU
- As operações de negociação no mercado foram a base para a geração do estado e do ambiente no qual o agente intera
- O agente utiliza uma rede neural sequencial com a seguinte estrutura de neurônios
  - 6 -> 64 -> 32 -> 3
  - Os parâmetros de entrada são: bid, ask, last, volume, flags, volume_real
  - Os parâmetros de saída são: mantêm, vende, compra
- A cada 256 iterações (operações no mercado financeiro) o agente refaz o treinamento, sendo o ajuste do parâmetro Q dado pela equação de bellman.
- Equação de Bellman
  - Relaciona o valor de estar em um estado com os valores dos estados futuros que se seguem. Isso permite que o agente faça previsões sobre o valor futuro esperado de suas ações, guiando-o na escolha de ações que maximizem a recompensa cumulativa ao longo do tempo.

Resultados





Considerações Finais

- Percebe-se que os resultados das negociações estão estabilizando e em algum momento convergindo para um valor positivo.
- Existem diversos estudos sobre a eficiência do mercados financeiros e a hipótese do mercado eficiente (HME) é uma das mais conhecidas. A HME afirma que os preços dos ativos refletem todas as informações disponíveis e, portanto, é impossível superar o mercado de forma consistente.
- No entanto, a HME é frequentemente contestada, e muitos estudos mostram que é possível superar o mercado de forma consistente.
- O agente de trading está aprendendo a tomar decisões com base nas experiências passadas, no entanto o custo computacional é alto, e o tempo de treinamento é longo.
- Ajustes no modelo de aprendizado por reforço podem ser necessários para melhorar a performance do modelo e consequentemente a eficiência do agente.
- O sistema de trading pode ser melhorado com a inclusão de mais entradas, como book de ofertas, notícias relevantes, indicadores técnicos, etc.
- Para uma aplicação real, é necessário considerar custos de transação, slippage, impostos. Além disso, é necessário considerar o risco de overfitting e a necessidade de validação do modelo.
- A implantação de um sistema de trading automatizado pode ter melhores resultados com o uso de maquinas virtuais.

Links/Contatos & Dúvidas?

- O código do projeto está hospedado no Github
  - https://github.com/MilianoJunior/daytrade
- Miliano
  - milianojunior39@gmail.com
- Álvaro
  - alvaropag@gmail.com