

**Fundação Getulio Vargas
Escola de Matemática Aplicada
Curso de Graduação em Matemática
Aplicada**

**Deep Reinforcement Learning aplicado ao
mercado de ações**

Lucas Almada Gabriel

Rio de Janeiro - Brasil
2021

Fundação Getulio Vargas
Escola de Matemática Aplicada
Curso de Graduação em Matemática
Aplicada

Deep Reinforcement Learning aplicado ao
mercado de ações

“Declaro ser o único autor do presente projeto de monografia que refere-se ao plano de trabalho a ser executado para continuidade da monografia e ressalto que não recorri a qualquer forma de colaboração ou auxílio de terceiros para realizá-lo a não ser nos casos e para os fins autorizados pelo professor orientador.”

Lucas Almada Gabriel

Rio de Janeiro - Brasil
2021

Fundação Getulio Vargas
Escola de Matemática Aplicada
Curso de Graduação em Matemática
Aplicada

Deep Reinforcement Learning aplicado ao
mercado de ações

“Projeto de Monografia apresentado à Escola de Matemática
Aplicada como requisito parcial para continuidade ao trabalho de
monografia.”

Aprovado em ____ de _____ de _____

Grau atribuído ao Projeto de Monografia: ____

Professor Orientador: Flávio Codeço Coelho
Escola de Matemática Aplicada
Fundação Getúlio Vargas

LUCAS ALMADA GABRIEL

“DEEP REINFORCEMENT LEARNING APLICADO AO MERCADO DE AÇÕES”

Trabalho de Conclusão de Curso - TCC apresentado ao Curso de Graduação em Matemática Aplicada da Escola de Matemática Aplicada para obtenção do grau de Bacharel em Matemática Aplicada.

Data da Defesa: 07/12/2020

ASSINATURA DOS MEMBROS DA BANCA EXAMINADORA



Flávio Codeço Coelho
Orientador



Moacyr Alvim Horta Barbosa da Silva
Membro



Elizabeth Wegner Karas
Membro

Nos termos da Lei nº 13.979 de 06/02/20 - DOU nº 27 de 07/02/20 e Portaria MEC nº 544 de 16/06/20 - DOU nº 114 de 17/06/20 que dispõem sobre a suspensão temporária das atividades acadêmicas presenciais e a utilização de recursos tecnológicos face ao COVID-19, as apresentações dos Trabalhos de Conclusão de Curso, de forma excepcional, serão realizadas de forma remota e síncrona, incluindo-se nessa modalidade membros da banca e discente.

Sumário

1	Introdução	4
2	Deep Reinforcement Learning: Overview	5
2.1	Introdução a Reinforcement learning	5
2.2	Trade off Exploration e Exploitation	7
2.3	Funções de valor	8
2.4	Temporal Difference Learning	9
2.4.1	Q - learning	9
2.5	Deep Q Networks	10
2.6	Critic-only, actor-only e actor-critic	11
2.6.1	A2C	11
2.6.2	PPO2	12
3	Resultados	13
3.1	Dados	13
3.2	Benchmark	13
3.3	Aplicação dos algoritmos	14
3.3.1	A2C	15
3.3.2	PPO2	17
3.4	Comparação dos resultados	21
4	Conclusão	23

1 Introdução

Usar algoritmos para entrar em posições na bolsa de valores é algo comum no mercado financeiro.

Conforme as novas formas de utilizar machine learning vão sendo melhoradas, automaticamente existem pesquisas para aplicar os conhecimentos no mercado de ações. Não seria diferente com Deep Reinforcement Learning como pode ser observado ao longo das seções.

A maioria dos artigos que tratam de trade quantitativo focam em classificação ou regressão para prever retornos. Porém, o que não é muito explorado nas pesquisas é a transformação das predições em posições de compra e venda.

Os modelos que serão apresentados não são tomados como base para predições, mas sim, como um ambiente completo de tomada de decisão, a partir de tentativa e erro.

No decorrer das seções serão explicadas a construção dos ambientes e em sequência os algoritmos utilizados, a saber: A2C, PPO2.

O que está por trás do modelo de encontrar posições diretas de compra e venda é o aprendizado por reforço. Na presente pesquisa será abordada a integração entre Reinforcement Learning e redes neurais. Exemplos de aplicações que obtiveram bons resultados podem ser citados: [1] e [2]

O trabalho está dividido da seguinte forma: Na seção 2, é apresentado um overview de Deep Reinforcement Learning. Começando com uma introdução, passando pelo trade off entre exploration e exploitation, funções de valor e discussão dos algoritmos. Na seção 3 são apresentados os resultados.

2 Deep Reinforcement Learning: Overview

Na presente seção será apresentada uma introdução a Deep Reinforcement Learning. Serão abordados os elementos que formam a teoria e os algoritmos.

2.1 Introdução a Reinforcement learning

Reinforcement Learning é um processo de aprendizado no qual o agente interage diretamente com o ambiente, objetivando aprender por tentativa e erro.[3][4][5].

Diferentemente do que é visto nos modelos tradicionais de machine learning, a ideia de aprendizado por reforço é que o agente chegue ao aprendizado sem a necessidade de inputs e outputs bem definidos. A diferença consiste na interação direta do agente, sendo motivado por retornos que recebe ao percorrer o ambiente. Pode-se que dizer que o agente é treinado para conseguir um bom comportamento.

O sistema de Reinforcement Learning apresenta alguns elementos, a saber: agente, política, reward, função de valor e modelo.[3]

A política mapeia os estados existentes no ambiente às ações tomadas pelo agente. Possui um valor que representa a probabilidade de tomar uma ação em um certo estado. Conforme o agente se torna mais experiente no ambiente, as políticas são alteradas, ou seja, as probabilidades de ações mudam para cada estado, até que se chegue em uma política ótima. Uma política ótima, é a política responsável por maximizar a expectativa de reward total.

Como dito em [3] a política é uma das principais propriedades de Reinforcement Learning, sendo suficiente para que o agente seja capaz de chegar a um comportamento.

O reward é o sinal entregue ao agente pelo ambiente em uma espécie de feedback sobre estados e ações tomadas. O objetivo final será a maximização do reward.

A forma de se chegar ao maior número possível é objeto de muitos estudos, os quais serão abordados. Reward acumulado com um fator de desconto $\gamma \in (0, 1)$ geram o retorno, definido como: $R_t = \sum \gamma^k r_{t+k}$

O desconto indica a importância temporal de cada reward. Preferências míopes apresentam um menor valor de gama, pois quanto menor, menos influência terão ações futuras. Esse fator aparece de forma interessante em aplicações de Finanças, pois pode ser pensando como forma de trazer o dinheiro a um valor presente.

A função de valor é responsável por calcular o valor final do reward esperado. É possível pensar em reward e função de valor, analogamente, como a estrutura conceitual de Planejadora e Fazedor narrada por Richard Thatcher no Livro Misbehaving [6]

Modelo é uma característica não obrigatória pelo fato de métodos model-free não necessitarem de modelos, uma vez que o agente treina por tentativa e erro. Em métodos model-based o modelo pode ser fornecido. A figura 1 ilustra os dois tipos de método.

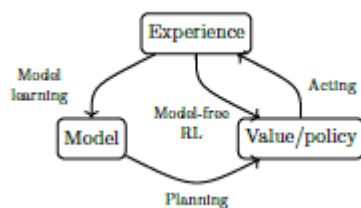


Figura 1: Model-free vs model-based

2.2 Trade off Exploration e Exploitation

No mercado de ações o investidor deve aprender a não sentir-se culpado por não acertar movimentos exatos dos preços. Ao investir, o agente pode deparar-se com a dúvida entre realizar uma posição vencedora ou esperar que ganhe mais com a movimentação do mercado na direção esperada.

Analogamente, o agente, ao percorrer o ambiente, deve escolher entre ficar em posições vencedoras já conhecidas ou explorar outros estados em busca de melhores recompensas, o que é conhecido como Trade off Exploration e Exploitation.

Uma das formas de tentar resolver esse dilema é através do algoritmo E-greedy, com $\epsilon \in (0, 1)$ e valor perto de 0. O agente mantém uma ação de valor alto com probabilidade $1 - \epsilon$ e explora o ambiente com probabilidade ϵ . Dessa forma o agente se beneficia um grande número de vezes de boas ações já conhecidas, sem deixar de testar outras possibilidades.[\[7\]](#)

2.3 Funções de valor

Funções de valor apresentam valor do estado para o agente. A qualidade desse valor é dada em termos de rewards futuros, ou ainda, reward esperado. Funções de valor são definidas para cada política adotada, utilizando a políticas definidas nos estados. A seguir como são definidas as funções de valor:

$$V : V^\pi \rightarrow \mathbb{R}, \quad V^\pi(s) = \mathbb{E}_\pi[R_t | s_t = s] = \mathbb{E}_\pi\left[\sum_{i=0}^{\infty} \gamma^i r_{t+i+1} | s_t = s\right]. \quad (1)$$

Funções de valor podem ser calculadas recursivamente com programação dinâmica. A equação 1 pode ser escrita da seguinte forma:

$$V^\pi(s) = \mathbb{E}_\pi[R_t | s_t = s] = \mathbb{E}_\pi\left[\sum_{i=0}^{\infty} \gamma^i r_{t+i+1} | s_t = s\right] = \mathbb{E}_\pi\left[r_{t+1} + \sum_{i=0}^{\infty} \gamma^i r_{t+i+2} | s_t = s\right]. \quad (2)$$

A equação 2 decomposta na equação de Bellman [8] tem o seguinte resultado:

$$\sum_a \pi(a|s) \sum_{s', r} p(s', r|s, a) [r + \gamma v_\pi(s')], \quad \forall s \in S. \quad (3)$$

A equação 3 apresenta todos os possíveis valores de a , r e s' (estado seguinte de s). Para os 3 valores é calculada a probabilidade, que é posta em cada círculo mostrado na figura. Todas as probabilidades são somadas resultando no valor esperado do estado.

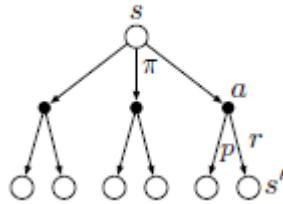


Figura 2: Esquema função de valor.

2.4 Temporal Difference Learning

Temporal-Difference Learning é um algoritmo que aprende sem a necessidade de modelos (model-free), tendo o aprendizado direto das experiências obtidas no ambiente.

Em TD, a função valor é atualizada em todas etapas, diferente de outros algoritmos que atualizam o valor apenas no final de todas as interações (bootstrap). A atualização é dada por:

$$V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]. \quad (4)$$

O α é o learning rate e $r + \gamma V(s') - V(s)$ o erro TD.

2.4.1 Q - learning

Q learning [8] é um algoritmo baseado em função de valor. Ele propõe achar a qualidade da ação em cada estado para que se obtenha a melhor recompensa futura.

É um dos algoritmos mais usados em Reinforcement Learning e base do Deep Q Network, algoritmo chave de Deep Reinforcement Learning. Q-Learning apresenta a seguinte definição de atualização:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \max_a Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)). \quad (5)$$

A função de valor em determinado estado é atualizada com o valor atual, reforço imediato e da diferença entre a máxima função de valor do estado seguinte e a função de valor atual. O objetivo em cada interação é selecionar a ação que maximize a função de valor do estado seguinte.

Ao ser executado, Q-Learning gera uma tabela (Q-table) que guarda os valores para cada ação-estado.

2.5 Deep Q Networks

Algoritmos tradicionais de RL são tabulares. O fato de valores estarem armazenados em tabelas gera custos que, para problemas maiores, se torna inviável. Uma forma de superar obstáculos é encontrar os valores armazenados por aproximações de funções. O algoritmo Deep Q Network [1] une Q-Learning e funções de aproximação, o que permite uma extração features relevantes a partir de inputs brutos. DQN usa a função de erro do Q-learning original para atualizar os parâmetros da rede neural:

$$(r + \gamma \max_{a'} Q(s', a', \theta_i^-) - Q(s, a, \theta_i))^2 \quad (6)$$

θ_i são os parâmetros da rede neural e θ_i^- são os parâmetros do target

2.6 Critic-only, actor-only e actor-critic

Os algoritmos apresentam diferenças, principalmente, no que se referem a estados e ações contínuas. No caso de ambientes de mercado financeiro essa diferença se dá principalmente no momento de compra e venda de ações.

- Critic-only

Nesse tipo de algoritmo são utilizados espaços discretos. É usada a função Q-valor para encontrar a política ótima que gere a maior recompensa. O fato de utilizar apenas espaços e ações discretas limita seu uso. Essa limitação é motivo pra não ser utilizado algoritmo dessa classe nas próximas seções. Q-Learning e Deep Q-learning são exemplos dessa abordagem.

- Actor-only

No actor-only o agente aprende a política ótima sozinho. Em vez da rede neural aprender o valor de Q, ela aprende diretamente a política. Exemplo desse tipo de algoritmo é o Policy Gradient (PG).

- Actor-critic

Nos algoritmos de actor-critic, existe a atualização da política e do valor simultaneamente, gerando tanto a otimização do valor quanto da política. Actor-critic permite a utilização de ambientes mais complexos como o do mercado de ações e por esse motivo será o escolhido para ser utilizado nas próximas seções. Esses algoritmos são: Advantage Actor-Critic (A2C) e Proximal Policy Optimization (PPO).

2.6.1 A2C

A ideia principal dos algoritmos A2C são as funções de vantagem (advantage function), utilizadas para reduzir a alta variância que as funções de valor possuem.

Definição das funções de vantagem:

$$A(s, a) = Q(s, a) - V(s) \tag{7}$$

Q: valor para a ação 'a' no estado 's'

V: valor médio do estado

A função de vantagem exprime a melhoria em relação ao valor médio da ação no determinado estado, ou seja, calcula a diferença de recompensa que a ação tomada tem em relação as outras.

2.6.2 PPO2

Algoritmo Proximal Policy Optimization é definido para não permitir que haja valores exagerados para a atualização da política. Nesse algoritmo é utilizado um limite de taxa entre políticas consecutivas.

A função que define o algoritmo é a seguinte [9]:

$$L^{CLI}(\theta) = \mathbb{E}_t\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} A_t\right] = \mathbb{E}_t[r_t(\theta) A_t]. \quad (8)$$

3 Resultados

Na presente seção serão discutidos os resultados dos modelos utilizados. Serão exibidos os benchmarks e feitas comparações entre resultados dos modelos e estratégias tradicionais de mercado financeiro.

3.1 Dados

Para as análises, foram utilizados OHLCV (abertura, máximo, mínimo, fechamento e volume dos preços) [10][11] retirados da base de dados do Yahoo Finance. O tempo do gráfico escolhido foi de 1 dia (1D) pelo fato de ser mais acessível uma vez que dados de mercado intraday apresentam preços altos. Foram consideradas as datas de 01/01/2010 até 31/12/2019.

3.2 Benchmark

Os algoritmos descritos na pesquisa serão comparados a alguns modelos de estratégia clássica do mercado financeiro:

1. Buy and Hold

Consiste em abrir uma posição no início no período e segurá-la até a liquidação sem nenhuma outra transação. Nos casos apresentados será considerada a compra da ação no começo do período de análise e venda no final.

2. Cruzamento de médias móveis

As médias móveis são as médias dos preços em um determinado período. O cruzamento das médias é uma estratégia utilizada no mercado e consiste em abrir posições no quando médias referentes a períodos diferentes se cruzam.

3. MACD

O MACD é um indicador muito utilizado por traders que operam baseado em análise técnica. Consiste na diferença entre duas médias móveis exponenciais. A utilização desse indicador se dá no cruzamento entre o sinal (média móvel de menor período) e a diferente entre as médias escolhidas como base.[12]

3.3 Aplicação dos algoritmos

Conforme exposto até aqui os algoritmos que buscaremos testar a performance são: A2C e PPO2

A escolha desses algoritmos se dá pelo fato de todos terem um espaço de escolhas de ações contínuo, uma vez que desejamos não apenas trazer a ação escolhida pelo agente, como também a quantidade realizada em cada ação.

A princípio será visto como o algoritmo performa de forma padrão. Com os hiperparâmetros mais usados e com as features padrão citadas. Nessa seção foram utilizados o papel da Apple e da IBM para avaliação.

A escolha dos refereridos papéis se deu pela alta liquidez, logo um menor valor de spread. Outra razão é como os papéis se comportam no período de teste, uma vez que as ações da Apple apresentam uma forte tendência de alta e da IBM uma lateralização.

- Período de treino: 01/01/2012 até 31/12/2017
- Período de teste: 01/01/2018 até 31/12/2019

3.3.1 A2C

IBM				Apple			
Best Day		13.68%		Best Day		5.89%	
Worst Day		-4.85%		Worst Day		-12.34%	
Best Month		24.73%		Best Month		17.46%	
Worst Month		-11.37%		Worst Month		-13.05%	
Best Year		12.68%		Best Year		66.55%	
Worst Year		-0.56%		Worst Year		-10.96%	
Avg. Drawdown		-5.01%		Avg. Drawdown		-3.59%	
Avg. Drawdown Days		116		Avg. Drawdown Days		23	

IBM						
[5 Worst Drawdowns]						
	Start	Valley	End	Days	Max Drawdown	99% Max Drawdown
1	2018-02-26	2019-02-01	2019-12-31	673	-27.81	-23.94
2	2018-01-11	2018-01-17	2018-01-23	12	-1.00	-0.92
3	2018-02-20	2018-02-20	2018-02-21	1	-0.57	0.00
4	2018-01-30	2018-01-31	2018-02-05	6	-0.30	-0.14
5	2018-02-15	2018-02-15	2018-02-16	1	-0.24	0.00

Apple						
[5 Worst Drawdowns]						
	Start	Valley	End	Days	Max Drawdown	99% Max Drawdown
1	2018-10-01	2018-12-27	2019-07-17	289	-34.84	-33.03
2	2019-12-18	2019-12-27	2019-12-31	13	-12.62	-12.34
3	2018-01-12	2018-01-31	2018-03-06	53	-11.04	-10.93
4	2019-07-26	2019-07-30	2019-10-03	69	-9.85	-8.70
5	2018-03-08	2018-04-19	2018-05-01	54	-9.62	-9.16

Figura 3: Relatório resultado A2C.

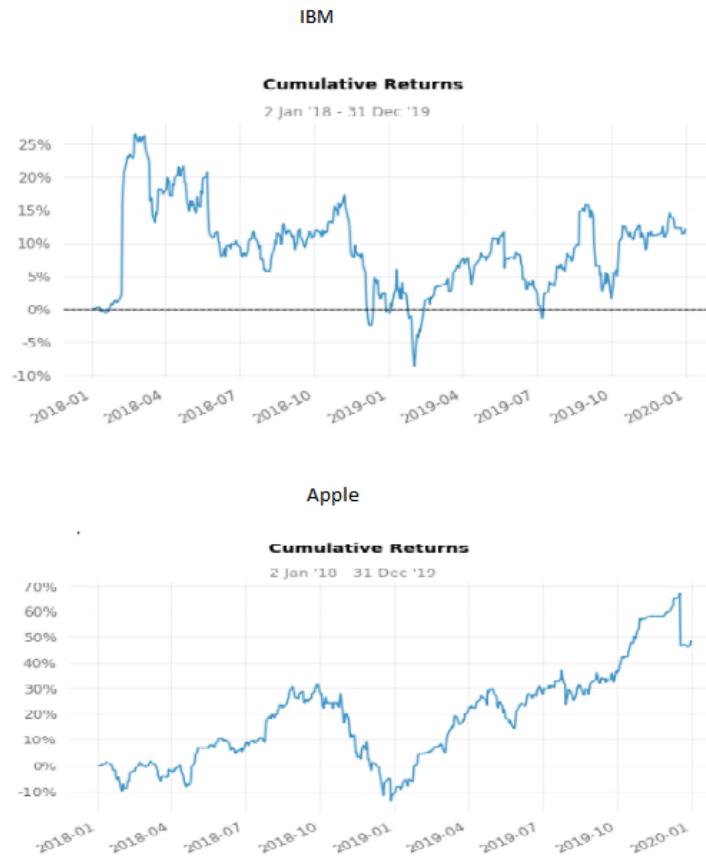


Figura 4: Retornos acumulados do modelo A2C.

3.3.2 PPO2

IBM				Apple			
Best Day		13.67%		Best Day		5.74%	
Worst Day		-5.57%		Worst Day		-9.26%	
Best Month		22.54%		Best Month		12.89%	
Worst Month		-7.16%		Worst Month		-22.6%	
Best Year		51.91%		Best Year		43.81%	
Worst Year		-10.07%		Worst Year		-7.9%	
Avg. Drawdown		-3.18%		Avg. Drawdown		-3.86%	
Avg. Drawdown Days		38		Avg. Drawdown Days		21	

IBM						
[5 Worst Drawdowns]						
	Start	Valley	End	Days	Max Drawdown	99% Max Drawdown
1	2019-01-07	2019-12-10	2019-12-31	358	-20.70	-19.03
2	2018-04-11	2018-05-24	2018-07-05	85	-10.52	-10.37
3	2018-09-24	2018-11-27	2018-12-18	85	-8.38	-8.32
4	2018-02-20	2018-03-01	2018-03-23	31	-3.69	-2.77
5	2018-07-17	2018-08-17	2018-08-23	37	-3.01	-2.91

Apple						
[5 Worst Drawdowns]						
	Start	Valley	End	Days	Max Drawdown	99% Max Drawdown
1	2019-09-12	2019-12-09	2019-12-31	110	-38.22	-36.38
2	2018-04-09	2018-05-15	2018-09-13	157	-17.94	-16.87
3	2018-12-07	2019-01-16	2019-04-18	132	-17.04	-16.87
4	2019-05-15	2019-06-03	2019-07-01	47	-4.62	-4.31
5	2019-08-14	2019-08-30	2019-09-10	27	-4.62	-4.56

Figura 5: Relatório resultado PPO2.

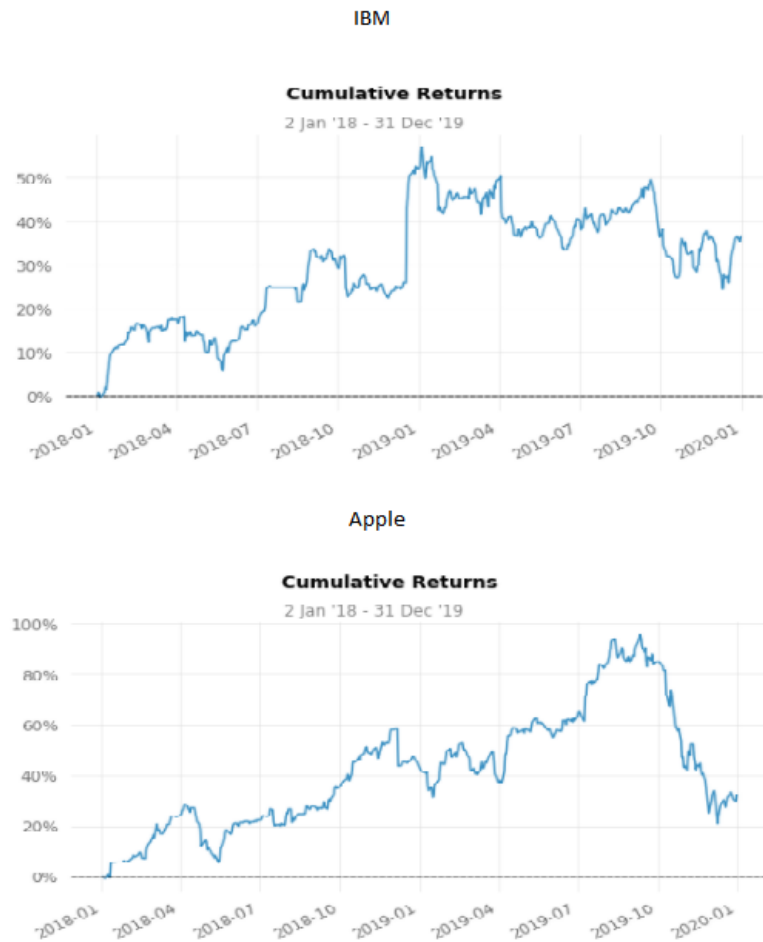


Figura 6: Retornos acumulados do modelo PPO2.

Como pode ser visto nas figuras 3 e 5 os modelos apresentam boa performance e poucos extremos. Em etapas posteriores podem ser limitadas as perdas, criando stops de forma que os retornos acumulados sejam ainda maiores com a diminuição das perdas. É comum que sejam usados benchmarks para testar a estratégia. Como feito em [11] e [10] será utilizada a estratégia de Buy and Hold, exercendo a compra no começo do período de teste e levando a posição até o final do tempo determinado.

Ambos os modelos performam melhor do que apenas comprar o papel da Apple em 2018. Porém, em 2019 como as ações da Apple apresentam uma forte tendência de alta, ter comprado no começo do ano e segurado a posição teria sido a melhor estratégia.

É verificado que para papéis com uma tendência o modelo não necessariamente segue a tendência principal. Na seção que é discutido o trade off entre exploration e exploitation, é explicado que o modelo eventualmente pode sair de uma posição ganhadora para tentar posições melhores. Uma maior taxa de exploração gera posições mais curtas, uma vez que a tendência é tentar melhores trades do que os atuais.

A seguir o caso de utilização da estratégia Buy and Hold para o mesmo período nas ações da Apple



Figura 7: Buy and Hold

Tão importantes quanto estabelecer ganhos é limitar perdas. Entender as perdas dos algoritmos é importante para gerenciamento de risco e para verificar o perfil do investidor que irá fazer a utilização da técnica. Alguns modelos podem apresentar elevados retornos mas permacerem bastante tempo em uma posição perdedora, gerando estresse ao investidor.

O modelo PPO2 apresenta uma média bem menor de dias de perdas como uma média de 38 dias de perdas para IBM enquanto A2C possui uma média de 116 dias, equivalente a 4 meses de perda em média, o que pode ser impensável para um investidor mais avesso a altos riscos.

3.4 Comparação dos resultados

Na subseção sobre benchmarks são apresentados os modelos clássicos utilizados para entrar em posições no mercado. A seguir, serão mostrados os gráficos dessas ações no período de teste, assim como os pontos de entrada baseado nos sistemas de benchmark.

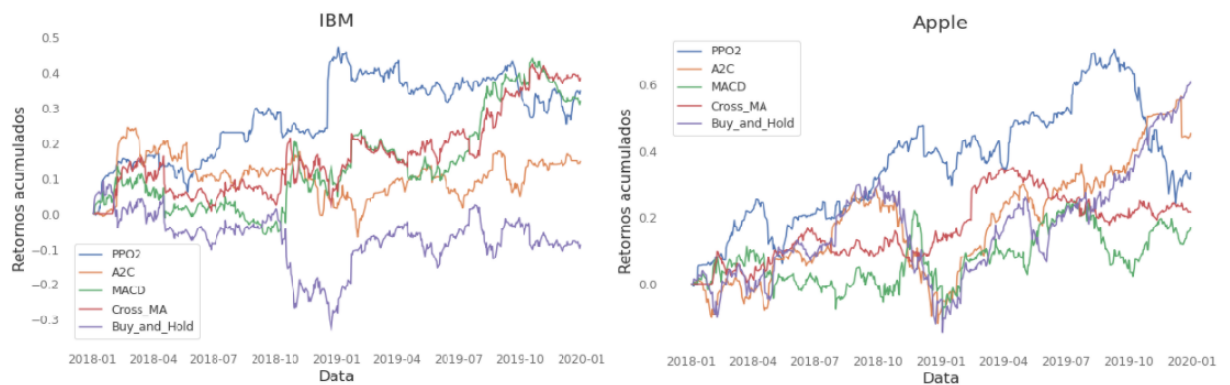


Figura 8: Comparação de Resultados

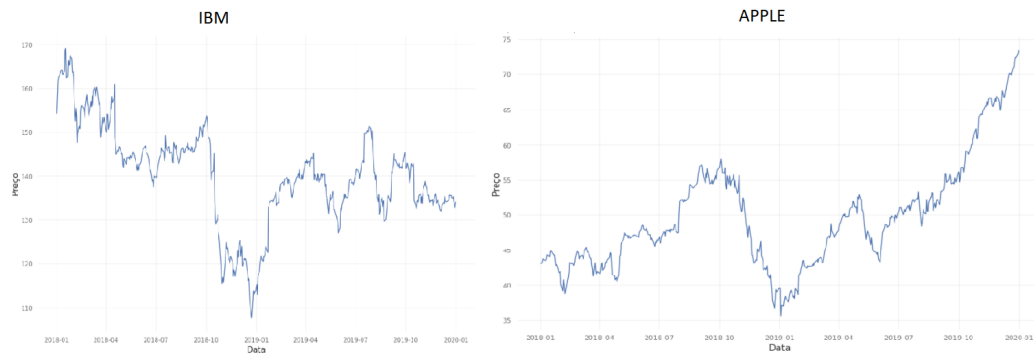


Figura 9: Gráfico de preços

Os resultados apresentados na figura 8 são a síntese do que o trabalho se propõe a apresentar. A figura 9 apresenta os gráficos dos preços da IBM e da Apple para comparação de performance dos modelos em diferentes estados do mercado.

Em relação ao gráfico dos modelos da IBM pode ser verificado que o algoritmo PPO2 tem uma performance bastante consistente. No final do processo ele tem ganhos acumulados um pouco menores que a estratégia de cruzamento de médias mas é o que apresenta maiores retornos na maior parte do tempo. Algo que o modelo também apresenta é o equilíbrio durante uma queda abrupta no preço da ação. No meses finais de 2018 há uma grande queda nas ações da IBM. O investidor que tivesse segurado a ação no Buy and Hold teria sofrido um grande prejuízo nessa época.

A estratégia de cruzamento de médias oscila entre ganhos e perdas podendo indicar uma indefinição. Já o PPO2 se mantém estável durante todo o período de queda. O A2C apresenta lucro mas com performance bem abaixo que as estratégias clássicas.

Nas ações da Apple, o PPO2 também mostra um maior ganho durante a maior parte do tempo mas no final é apenas o terceiro maior retorno. O A2C performa melhor que o PPO2 em relação a retornos totais. Embora tenha um resultado final melhor, o A2C tem um rendimento abaixo quase todo o processo. Levando em conta que o tempo de saída não é conhecido e poderia ser realizado em qualquer parte do range testado, na maioria do tempo teria sido melhor usar o PPO2.

4 Conclusão

Muitos algoritmos que possuem boa performance em diversos campos da literatura, tão logo são utilizados, recebem atenção do mercado financeiro.

Prever com uma boa taxa de acerto o movimento das ações é o sonho de muitos investidores e utilizar os melhores modelos conhecidos é uma das formas de encontrar essa performance desejada.

Não é diferente em relação aos algoritmos de Reinforcement Learning. Como pode ser observado em [12], existem alguns artigos com abordagem de trading e utilização de Reinforcement Learning. Porém, a quantidade de artigos sobre o assunto é bem baixa se comparado a modelos criados com algoritmos clássicos de machine learning.

A baixa menção ao assunto reflete em sites e blogs que passam informações sobre análise de dados, uma vez que existem poucas postagens sobre o assunto.

Por ser ainda um terreno pouco explorado, existe muito o que se estudar e melhorar em Reinforcement Learning. Em relação ao presente trabalho existem melhorias que serão feitas.

A primeira adição futura é uma melhor engenharia de features. As features usadas seguem o modelo padrão de outros artigos ao utilizar OHLC das ações como input para o modelo. Uma boa modelagem tem a capacidade de aumentar significativamente a performance de modelos e algumas ideias de features serão implementadas futuramente.

Outra melhoria a ser feita é a estruturação do ambiente construído. O ambiente é o cerne do funcionamento do modelo, e algumas ideias relacionados aos passos a serem tomadas pelo agente entre as posições ainda precisam ser implementadas.

Por fim, uma mudança que será feita e segue uma linha de mudança de ambiente, é permitir que sejam colocadas mais ações em um mesmo modelo. Essa melhoria segue a linha de adicionar um portfólio, ou seja, adicionar várias ações e deixar que o agente escolha as oportunidades. No modelo atual, como é utilizada uma ação por modelo, o agente escolhe apenas que posição tomar em um determinado período.

Referências

- [1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dhharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015.
- [2] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, jan 2016.
- [3] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.
- [4] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.
- [5] Yuxi Li. Deep reinforcement learning: An overview. *CoRR*, abs/1701.07274, 2017.
- [6] Ananish Chaudhuri. Misbehaving: The making of behavioral economics by richard h. thaler, w.w. norton and company, new york, 2015, xvi + 415 pp., hardcover, usd 27.95, isbn: 978-0-393-08094-0. *Journal of Behavioral and Experimental Economics (formerly The Journal of Socio-Economics)*, 60(C):64–65, 2016.
- [7] Matthew E. Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(1):1633–1685, 2009.
- [8] Christopher John Cornish Hellaby Watkins. *Learning from Delayed Rewards*. PhD thesis, King’s College, Cambridge, UK, May 1989.

- [9] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.
- [10] Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deep reinforcement learning for trading, 2019.
- [11] Chien Yi Huang. Financial trading as a game: A deep reinforcement learning approach, 2018.
- [12] Ansari Saleh Ahmar. Sutte indicator: an approach to predict the direction of stock market movements, 2019.