AGENTES AUTÔNOMOS E REINFORCEMENT LEARNING

ELETIVA

Projeto Final - Entrega Intermediária

Caio Emmanuel

Entrega intermediária contendo três casos de uso e aplicação de agentes autônomos e escolha do meu caso de entrega.

São Paulo

Junho de 2022

Sumário

1 Mercado Financeiro

1.1 Ambiente

O *environment* para esse caso de uso consiste em ambientes capazes de simular as operações do mercado financeiro de compra e venda, além de ser capaz de gerar - ou ler de uma base externa - dados de preço com o formato dos mercados mais tradicionais (e.g. mercado acionário). Para esse caso de uso, vamos usar exemplo o ambiente *AnyTrading*[1].

O *AnyTrading* é uma coleção de ambientes para treinamento de agentes autônomos desenvolvida sobre a biblioteca *Gym*[2] da *OpenAI*. A biblioteca já é responsável por gerar um estado aleatório com dados de preço sobre um ativo financeiro.

1.2 Agente

Um agente para esse ambiente atua como um *trader* no mercado tradicional, podendo executar as seguintes ações:

- *Buy* = 1: comprar um ativo;
- *Sell* = 0: vender um ativo.

E seu objetivo é maximizar o retorno ao final de uma *window* de amostragem do preço do instrumento analisado.

1.3 Recompensas

A recompensa no ambiente utilizado como exemplo é o retorno da estratégia do agente ao longo do tempo, mas poderia ser a diferença entre esse e uma *baseline* mais simples (e.g. *Buy Hold*), ou, se quisessemos fazer um agente que acerte a direção apenas, um reforço positivo para toda vez que acertar a posição.

1.4 Trabalhos correlatos

Existem diversos trabalhos na área de finanças quantitativas. Várias soluções foram desenvolvidas para lidar com este problema, no campo da econometria, alguns dos modelos mais populares são *auto-regressive method (AR), moving average (MA)* e *autoregressive integrated moving average (ARIMA)*[3,

4] que, explicando brevemente, inferem o preço em um momento t utilizando uma combinação linear dos preços nos instantes $\{1,2,...,t-1\}$. Entretanto, um problema desta classe de modelos é que eles partem de premissas sobre a diferença entre o preço em dois instantes (como distribuição-t ou variáveis independentes e identicamente distribuídas), que não se confirmam com dados reais na maioria das vezes.

Já no campo de *soft computing* (área que envolve inteligência artificial), algumas das principais soluções incluem a utilização de *Redes Neurais Artificiais*[5] e *Support Vector Machine*[6, 7] e outras da área de *Deep Learning*[8], que têm atraído olhares devido à capacidade dessa estrutura de extrair *features* abstratas dos dados (inclusive de dados não estruturais, como *tweets* ou *headlines* de jornais[9, 10]). Entretanto, essa classe de modelos têm uma série de problemas que dificultam a implementação como custo de treinamento das redes neurais, expertise necessária para lidar com as especificidades e restrições dos diferentes tipos de dados que podem ser passados por estas.

Portanto, é provável que um agente treinado sobre esse ambiente, sem um *hardware* de alta performance e sem o rigor de ser construído por um time de *experts* na área, dificilmente irá performar melhor que a maioria dos outros trabalhos na área ou melhor do que o exemplo na própria documentação do ambiente.

2 Jogo *Doom*

2.1 Ambiente

O exemplo de *environment* para esse caso de uso é o *ViZDoom*[11], um ambiente construído sobre o *ZDoom*[12] para integrá-lo com o *gym*.

O ambiente emula perfeitamente o cenário do jogo *Doom* e sua API foi criada especificamente para trabalhar com agente autônomos, como citado na documentação.

2.2 Agente

Um agente para esse ambiente atua como um jogador regular, o espaço de observação são os *frames* do jogo, e o agente pode executar as seguintes ações:

- *left* = [1,0,0]: andar para a esquerda;
- *right* = [0,1,0]: andar para a direita;
- *shoot* = [0,0,1]: atirar.

2.3 Recompensas

Toda ação produz uma recompensa para o agente, sendo:

- *reward* = -6: caso atire e erre;
- *reward* = 100: caso atire e mate um monstro;
- *reward* = -1: caso nenhum dos anteriores, para evitar que fique parado ou repetindo um círculo, por exemplo

2.4 Trabalhos correlatos

Os principais trabalhos na área são dos próprios criadores do jogo[13, 14], desenvolvendo um agente com uma abordagem de *Deep Q-Learning*. E os resultados são promissores, como podem ser vistos na Figura 1 e Figura 2.

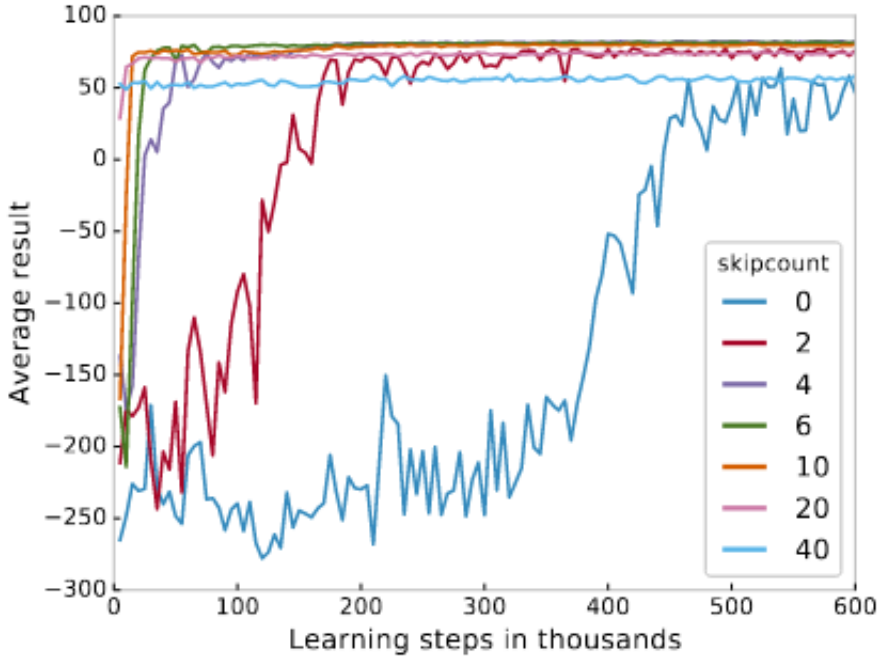

Figura 1: Average Result per Steps - Doom Agent

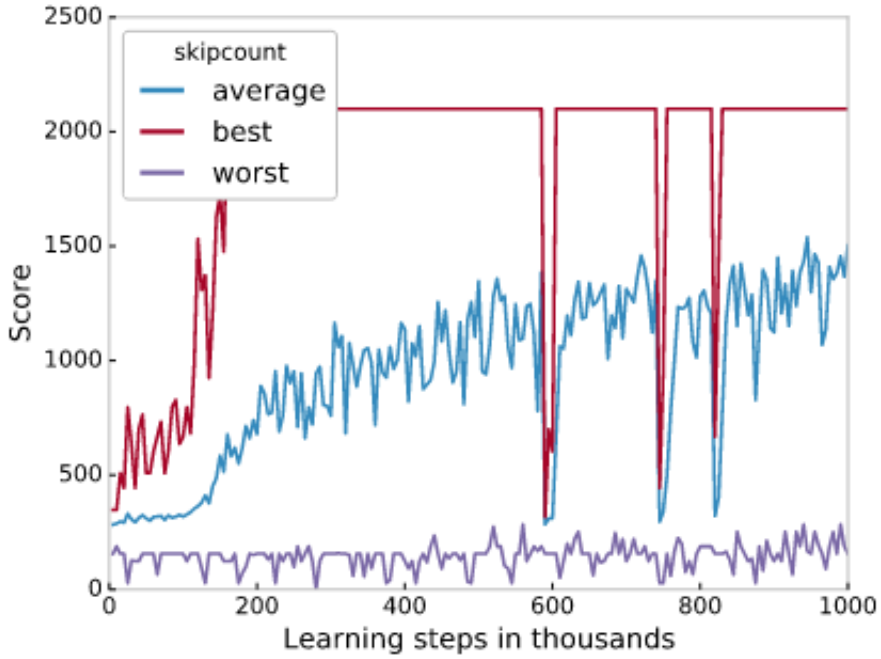

Figura 2: Score per Steps - Doom Agent

3 Biomecânica

3.1 Ambiente

O exemplo de *environment* para esse caso de uso é o *OpenSIM RL*[15], uma biblioteca construída para simular algums movimentos mecânicos do corpo, como andar ou mover as mãos.

3.2 Agente

Um agente para esse ambiente tenta aprender alguma das funções fisiológicas regulares, como correr ou mexer os dedos, o espaço de observação é um vetor com os valores de tensão e força nos músculos, juntas e dados sobre velocidade e aceleração do corpo. As ações são também um vetor de 0's (zeros) e 1's (uns) para "liberar"ou "estressar"um músculo.

3.3 Recompensas

As recompensas variam a depender do ambiente escolhido, mas a recompensa para o mais popular, o *L2RunEnvPermalink*, que visa criar um agente que aprenda a correr, tem a recompensa dada pela distância alcançada.

3.4 Trabalhos correlatos

A maioria dos trabalhos sobre esse ambiente foram expostos na *NeurIPS*, uma das maiores conferências do mundo na área de aprendizado de máquina e neurociência, realizada anualmente até o ano de 2019 e após isso congelada até 2022 por conta da pandemia.

Os resultados são promissores e podem ser vistos na documentação nas referências, já que os resultados são vídeos ou animações do agente em ação.

4 Minha escolha

Meu método de escolha é baseado na praticidade do ambiente e para isso eu considero: *setup* necessário para simular o ambiente e complexidade para receber o espaço de observação e enviar uma ação.

Dado isso, o ambiente do *OpenSIM* foi eliminado, pois a maioria dos ambientes estão quebrados e tomariam muito tempo para serem configurados devido a um abandono de ambientes mais antigos por parte dos desenvolvedores, que priorizam os ambientes das próximas competições.

Portanto, pensando nas aplicações potenciais de cada agente, eu decidi seguir com o *VizDoom*, por já possuir trabalhos acadêmicos na área, permitindo fazer um comparativo de desempenho.

Referências

[1] *gym-anytrading*. https://github.com/AminHP/gym-anytrading. Accessed: 2022-05-31.

[2] *gym-library*. https://www.gymlibrary.ml/. Accessed: 2022-05-31.

[3] J. D. Hamilton. *Time Series Analysis*. Vol. 2. Princeton University Press, 1994.

[4] Jeffrey Marc Wooldridge. *Introductory Econometrics: A Modern Approach*. ISE - International Student Edition. South-Western, 2009. ISBN: 9780324581621.

[5] Yakup Kara, Melek Acar Boyacioglu e Ömer Kaan Baykan. "Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange". Em: *Expert Systems with Applications* 38.5 (2011), pp. 5311–5319. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2010.10.027. URL: https://www.sciencedirect.com/science/article/pii/S0957417410011711.

[6] Alaa Sheta, Sara Ahmed e Hossam Faris. "A Comparison between Regression, Artificial Neural Networks and Support Vector Machines for Predicting Stock Market Index". Em: *International Journal of Advanced Research in Artificial Intelligence* 4 (jul. de 2015), pp. 55–63. DOI: 10.14569/IJARAI.2015.040710.

[7] Wei Huang, Yoshiteru Nakamori e Shou-Yang Wang. "Forecasting stock market movement direction with support vector machine". Em: *Computers Operations Research* 32.10 (2005). Applications of Neural Networks, pp. 2513–2522. ISSN: 0305-0548. DOI: https://doi.org/10.1016/j.cor.2004.03.016. URL: https://www.sciencedirect.com/science/article/pii/S0305054804000681.

[8] Ian Goodfellow, Yoshua Bengio e Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[9] Johan Bollen, Huina Mao e Xiaojun Zeng. "Twitter mood predicts the stock market". Em: *Journal of Computational Science* 2.1 (2011), pp. 1–8. ISSN: 1877-7503. DOI: https://doi.org/10.1016/j.jocs.2010.12.007. URL: https://www.sciencedirect.com/science/article/pii/S187775031100007X.

[10] László Nemes e Attila Kiss. "Prediction of stock values changes using sentiment analysis of stock news headlines". Em: *Journal of Information and Telecommunication* 1 (2021), pp. 1–20. DOI: 10.1080/24751839.2021.1874252.

[11] *ViZDoom*. https://github.com/mwydmuch/ViZDoom. Accessed: 2022-05-31.

[12] *ZDoom*. https://github.com/rheit/zdoom. Accessed: 2022-05-31.

[13] Marek Wydmuch, Michał Kempka e Wojciech Jaśkowski. *ViZDoom Competitions: Playing Doom from Pixels*. 2018. DOI: 10.48550/ARXIV.1809.03470. URL: https://arxiv.org/abs/1809.03470.

[14] Michał Kempka et al. "ViZDoom: A Doom-based AI Research Platform for Visual Reinforcement Learning". Em: (2016). DOI: 10.48550/ARXIV.1605.02097. URL: https://arxiv.org/abs/1605.02097.

[15] *OpenSIM RL*. http://osim-rl.kidzinski.com/docs/home/. Accessed: 2022-05-31.