CONTROLE DE UM ROBÔ MÓVEL COM PÊNDULO INVERTIDO POR MEIO DE APRENDIZADO POR REFORÇO

Miguel Lopes de Moraes, Danilo Hernani Perico
Centro Universitário FEI,
São Bernardo do Campo, SP, Brasil
e-mail: miguellopes54954@gmail.com , dperico@fei.edu.br

Resumo: Este artigo apresenta o conceito de aprendizado por reforço aplicado ao problema do *CartPole* (pêndulo invertido preso a um robô móvel) e, nesse problema o robô deve entender o ambiente em que está e utilizar de suas ações para não deixar o pêndulo cair. Os métodos de aprendizado por reforço usados nesse projeto foram o Q-Learning e o Deep Q Network (DQN). Foi possível ver pelos resultados que a utilização de ambos os métodos soluciona o problema, porém o DQN apresentou melhores resultados.

1. Introdução

Nesse trabalho será explorado o conceito de Aprendizado por Reforço, utilizando o algoritmo clássico Q-Learning que trabalha por meio de ações e recompensas para permitir que um agente entre em um ambiente desconhecido e aprenda qual a melhor ação a ser tomada em cada estado do domínio.[3]

Da mesma forma, será utilizado também o conceito de *Deep Reinforcement Learning – DRL* (Aprendizado por Reforço Profundo), que utiliza uma Rede Neural Artificial como base para extrair mais conhecimento domínio e permitir que o agente possa adquirir mais conhecimento. De forma geral, no DRL a Rede Neural aprende por meio do reforço de experiências individuais geradas pelo agente. Assim, para a implementação deste método, foi utilizado o algoritmo Deep Q Network (DQN).[4]

Portanto, o objetivo desse projeto é o estudo e a implementação de técnicas de Aprendizado por Reforço e Aprendizado por Reforço Profundo para realizar o controle de posição de um pêndulo invertido em base móvel, sendo que a base é um robô com rodas.

Assim, o domínio aplicado a essa proposta é um ambiente criado no simulador Webots com o problema completo do *CartPole*, onde a base móvel do sistema é um robô com tração nas quatro rodas.

2. Fundamentação Teórica

O Q–Learning é um dos algoritmos de Aprendizado por Reforço. Ele foi desenvolvido por Watkins (1989) como uma forma de buscar um aprendizado capaz de entender e atuar sem um conhecimento prévio do ambiente [1]. Vale ressaltar dois conceitos fundamentais desse algoritmo, que foram explorados ao decorrer desse projeto, o primeiro é a utilização de variáveis na construção de estados e o segundo e a taxa de exploração (ε – *Greedy*).

Já o *Deep Q Network - DQN* é uma técnica de aprendizado de máquina que utiliza de uma Rede Neural Convolucional com o Aprendizado por Reforço. A rede é responsável pela tomada de decisões e o aprendizado por reforço faz a avaliação da ação executada por meio das recompensas [2]

3. Metodologia

Inicialmente, para a simulação do robô, já com o sistema de *CartPole*, foi criado um programa em Python, que envolve o método Aprendizado por Reforço. Por meio do Webots foi possível visualizar o problema do pêndulo invertido com base móvel robótica e a solução dada por esse método.

Para a implementação dos métodos citados anteriormente, foram desenvolvidos códigos com base no que foi descrito. O código que utilizou o Q-Learning utilizou como base a biblioteca OpenAI para realizar a captura das variáveis e o equacionamento do Q-Learning para o treinamento do agente. Já para o DQN, o código utilizado teve como base um projeto que utiliza a biblioteca TensorFlow para a criação de uma Rede Neural e o Q-Learning para a captura de variáveis.

Para o algoritmo com Q-Learning foram feitos 3 experimentos, variando na construção dos estados e na taxa de exploração (ε – *Greedy)*.

Assim, foram feitas 30 repetições de cinco mil episódios para cada experimento. Ao término desse processo foi feito a média móvel dos resultados gerados, para assim formar comparações entre os algoritmos.

Cada resultado gerou um gráfico que contém os episódios na horizontal e as recompensas na vertical, essa recompensa é média móvel das trinta experiencias realizadas.

4. Resultados

Para uma melhor compreensão da aprendizagem, a Figura 1 demonstra um caso em que o robô realiza apenas ações aleatórias, ou seja, não há nenhum modo de aprendizagem, é possível ver no gráfico que as recompensas são praticamente as mesmas e sua curva permanece estável. Portanto, a curva deve manter um crescimento em suas recompensas para demonstrar a aplicação do aprendizado por reforço no problema, neste caso, por exemplo, os valores estão sempre em uma média comum, não crescendo e variando com o curso dos episódios.

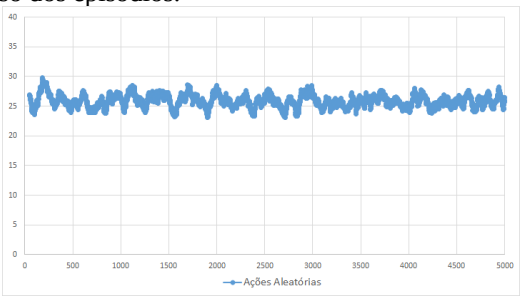

Figura 1 – Gráfico demonstrando apenas Ações aleatórias

A Figura 2 compara, em cinco mil episódios, todos os métodos. A curva em azul representa as ações aleatórias, a curva roxa o código Q-Learning com 4 variáveis na composição do estado (posição do pêndulo θ, velocidade angular do pêndulo ω, posição do robô X, velocidade do robô V), a verde com duas variáveis (posição do pêndulo θ, velocidade angular do pêndulo ω) sem desconto na taxa de exploração, em vermelho com desconto e finalmente em azul claro o uso do DQN.

É possível notar a diferença de todos com o das ações aleatórias, por todas elas ter uma curva de recompensa crescente, demonstrando suas diferenças pelas recompensas alcançadas.

Nesse gráfico é possível comparar as diferenças entre os tipos de usos do código com apenas o Q-Learning, tais como a diferença de resultados com o uso da taxa de exploração, demonstrado tanto pela curva vermelha quanto pela verde, dessa forma pode-se notar neste problema que quanto mais o robô explora, menos aprende, pois com o passar dos episódios ele terá uma base de estados de recompensa. Também é possível notar a diferença de velocidade de aprendizado dependo da quantidade de variáveis na formação dos estados, representados pela curva roxa, verde e vermelha, podendo presumir que quanto maior a quantidade de variáveis maior o custo operacional do código.

Destarte, vale ressaltar o uso do algoritmo DQN, este demonstrou uma eficácia no aprendizado, como pode ser visto pela curva em azul claro, a mesma conseguiu obter bons resultados durante os episódios, usando o método de aprender com ações passadas.

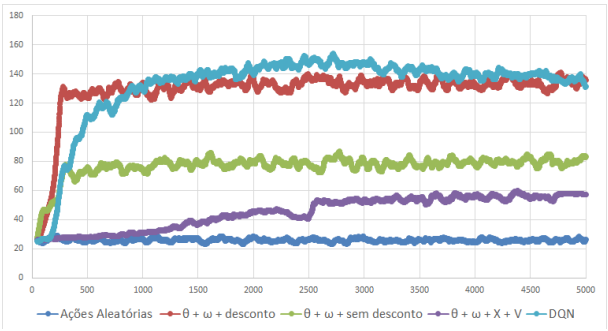

Figura 2 – Gráfico demonstrando o resultado da comparação dos métodos Q--Learning e DQN.

Assim, o terceiro experimento apresenta uma comparação entre o código que possui o DQN (curva azul) e o código anterior que possui apenas o Q–Learning com o estado contendo θ e ω juntamente com decaimento na taxa de exploração (curva vermelha). A Figura 3 apresenta o resultado do experimento.

Analisando o comportamento de ambas as curvas, é possível notar um crescimento inicial de recompensas e a partir de um dado momento uma estabilidade ou decaimento da curva, logo possível assumir que ambos os códigos buscaram estabilidade. Ao analisar as barras de erro é notável uma diminuição, assim é presumível entender esse comportamento, dado que o mesmo busca uma maior estabilidade.

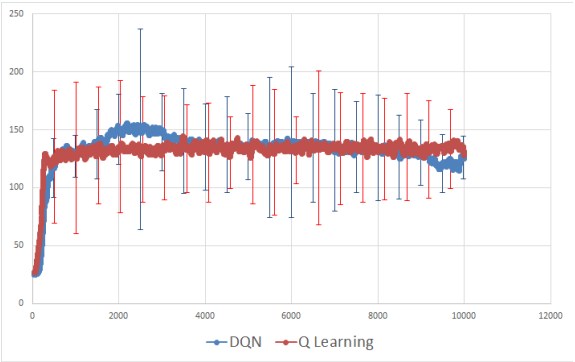

Figura 3 – Título da figura.

5. Conclusões

Pela análise do que foi discutido durante esse relatório, é possível concluir que a taxa de exploração(ε – *Greedy)* do aprendizado por reforço é um conceito pequeno, porém muito importante para o acúmulo de experiências do programa, como pode ser visto na seção 4, dentro dos experimentos feitos, o robô atuou de melhor forma quando sua exploração era muito baixa, fazendo com que ele apenas seguisse com o que aprendeu, entretanto ao aumentar sua exploração o mesmo continuou com sua dinâmica, sem alcançar bons resultados em seu objetivo final. Da mesma forma, foi possível analisar a diferença no número de variáveis em relação ao tempo, como pode ser visto durante o gráfico da Figura 2 da seção 4, quanto maior o número de variáveis, maior o tempo de episódios para o algoritmo atingir seu objetivo.

Da mesma forma, foi possível observar o comportamento de barras de erro na Figura 3 proposto pela seção 4, no início dos episódios as barras tendem a crescer devido ao aumento pela busca por boas recompensas, mas com a passagem dos episódios há uma diminuição, demonstrando o aprendizado da máquina.

Por fim, pode-se concluir que o uso do método de aprendizado por reforço resolve o problema do pêndulo invertido.

6. Referências

[1] D. H. Perico o de heurísticas obtidas por meio de demonstrações para aceleração do aprendizado por reforço. Centro Universitário da FEI, São Bernardo do Campo, 2012.

[2] TAMADA, Vítor Kei Taira. Estudo de caso de Deep Q-Learning. Citations on, p. 74, 2019.

[3] WATKINS, C. J. C. H. Learning from Delayed Rewards. 1989. Tese (Doutorado) – King's College, Oxford.

[4] SEWAK, Mohit. Deep Q Network (DQN), Double DQN, and Dueling DQN. In: DEEP Reinforcement Learning: Frontiers of Artificial Intelligence. Singapore: Springer Singapore, 2019. P. 95–108. ISBN 978-981-13-8285-7