 <p>INSTITUTO FEDERAL ESPIRITO SANTO Campus Serra</p>	<p style="text-align: center;">IFES</p> <p style="text-align: center;">Inteligência Artificial</p> <p style="text-align: center;">Exercício de Programação 3: Aprendizado por Reforço (GYM + Stable Baselines)</p>	<p style="text-align: center;">Nota</p>
--	--	---

Professor: Sérgio Nery Simões

Data: 27/06/2024

Nome: _____

Turma: _____

Informações:

- O trabalho deve ser submetido até a data limite na atividade do AVA.
- Deve ser submetido um arquivo compactado contendo o código em Python dos programas desenvolvidos e o relatório em formato pdf.
- É permitido que os alunos conversem e discutam suas soluções, mas os códigos-fonte devem ser produzidos individualmente. Se forem identificados casos de cola, serão abertos processos que podem culminar no desligamento do aluno.

Descrição do Trabalho

1) Introdução

O aprendizado por reforço (RL – *Reinforcement Learning*) tem se destacado como uma área fundamental da inteligência artificial, permitindo que agentes autônomos aprendam a tomar decisões e aprimorem seu desempenho através da interação com o ambiente. É um paradigma de aprendizado em que um agente aprende a tomar ações em um ambiente dinâmico, visando maximizar uma medida de desempenho chamada recompensa. Ao interagir com o ambiente, o agente recebe *feedback* na forma de recompensas positivas ou negativas, o que permite que ele aprenda a realizar ações que levem a solução do problema.

Uma das principais vantagens do aprendizado por reforço é sua capacidade de lidar com problemas complexos, nos quais a solução requer uma sequência de ações e não existe uma resposta correta predefinida. Ao aprender por meio da interação com o ambiente, o agente pode descobrir estratégias eficientes e adaptar seu comportamento de acordo com as mudanças nas condições do ambiente.

Além disso, o aprendizado por reforço tem inúmeras aplicações práticas em diferentes áreas, como robótica, jogos, controle de processos, otimização e muitas outras. Através do treinamento de agentes capazes de tomar decisões autônomas, o aprendizado por reforço tem o potencial de impulsionar avanços significativos em diversos campos, tornando-se uma área de pesquisa e desenvolvimento promissora.

Este trabalho tem como objetivo abordar o tema do aprendizado por reforço aplicado ao treinamento de agentes para resolver desafios nos ambientes “*Mountain Car*” e “*Car Racing*” disponíveis na biblioteca GYMNASIUM (que contém diversos ambientes didáticos). Por meio da utilização da biblioteca Stable Baselines, exploraremos a aplicação de diversas políticas para solucionar problemas por aprendizado por reforço em dois ambientes diferentes.

Problema 1: aplicação de Aprendizado por Reforço no ambiente “*Mountain Car*”

Descrição do Problema

O problema “*Mountain Car*” é um ambiente de aprendizado por reforço onde um carro deve superar a força da gravidade para alcançar o topo de uma montanha. O objetivo do agente é fazer com que o carro suba a montanha dentro de um limite de tempo, usando uma quantidade limitada de força disponível.

O ambiente é representado por uma pista de corrida bidimensional, que consiste em uma colina com uma forma peculiar. A colina possui um vale estreito e achatado, onde o carro começa sua jornada, e duas montanhas de cada lado do vale. O carro é inicialmente colocado no vale e deve acelerar ou frear para ganhar impulso e superar a gravidade, tentando atingir o topo de uma das montanhas.

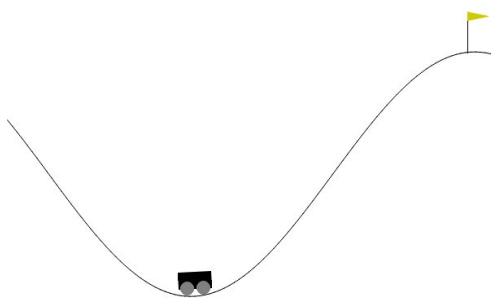


Figura 1: ambiente “*Mountain Car*”

A física do problema é definida por um conjunto de regras e leis que governam o movimento do carro. O carro é modelado como um objeto com massa e está sujeito a duas forças principais: a força gravitacional, que puxa o carro para baixo, e a força do motor, que pode acelerar ou frear o carro. No entanto, o carro possui uma limitação: sua força de motor não é suficiente para diretamente vencer a gravidade e subir a montanha íngreme.

Para superar esse desafio, o agente precisa usar estratégias inteligentes. Por exemplo, o agente pode acelerar o carro quando ele está descendo uma montanha para ganhar velocidade e *momentum*, o que pode ajudar a empurrá-lo para cima da próxima montanha. No entanto, o agente também precisa ter cuidado para não acelerar demais e perder o controle do carro ou ultrapassar a montanha.

O ambiente fornece observações ao agente, que incluem a posição do carro e sua velocidade atual. Além disso, o ambiente define regras de terminação, como um limite máximo de tempo ou uma posição específica que o carro deve atingir para considerar a tarefa concluída com sucesso. O agente recebe recompensas positivas quando o carro se move em direção ao topo da montanha e recompensas negativas quando o carro se move para trás ou quando o tempo limite é atingido.

Resolver o problema “*Mountain Car*” requer uma estratégia inteligente para equilibrar o uso adequado da força do motor, o gerenciamento de *momentum* e a exploração do ambiente. Os algoritmos de aprendizado por reforço são utilizados para treinar um agente capaz de aprender essas estratégias através da interação com o ambiente, permitindo que o carro supere os desafios físicos e alcance o topo da montanha dentro do limite de tempo estabelecido. Além disso, é necessário descobrir a política apropriada para solucionar o problema, já que nem todas podem ser aplicadas.

Especificações do Trabalho

1. Ambiente: “*Mountain Car*” disponível no link abaixo. É importante familiarizar-se com as características desse ambiente, incluindo ações disponíveis, observações, recompensas e condições de término.

https://gymnasium.farama.org/environments/classic_control/mountain_car/

2. Implementação: deve ser utilizada a biblioteca ***Stable Baselines*** para implementar o algoritmo de aprendizado por reforço. Essa biblioteca fornece uma série de algoritmos bastante utilizados, como DQN, PPO, A2C, entre outros. Você deve escolher um desses algoritmos e implementá-lo para resolver o ambiente “*Mountain Car*”.

3. Treinamento: uma vez escolhido o algoritmo, o agente deve ser treinado utilizando o algoritmo escolhido para solucionar o ambiente. O treinamento deve ser executado por um número suficiente de episódios para permitir que o agente aprenda a superar os desafios do ambiente.

Problema 2: aplicação de aprendizado por reforço no ambiente "Car Racing"

Descrição do Problema

Similar ao problema 1, porém o ambiente é 2D. O objetivo deste trabalho é aplicar técnicas de aprendizado por reforço para treinar um agente capaz de resolver o ambiente "Car Racing" disponível na biblioteca GYM. Esse ambiente simula uma pista de corrida onde um carro deve aprender a navegar pela pista e completar voltas no menor tempo possível, evitando colisões com obstáculos.

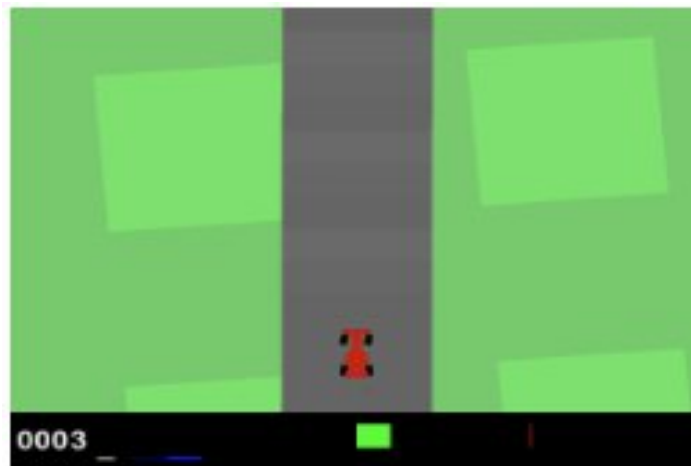


Figura 2: ambiente "Car Racing"

Especificações do Trabalho

1. Ambiente: o ambiente a ser utilizado é o "Car Racing" disponível no link abaixo. É importante familiarizar-se com as características desse ambiente, incluindo ações disponíveis, observações, recompensas e condições de término.

https://gymnasium.farama.org/environments/box2d/car_racing/

2. **Implementação:** deve ser utilizada a biblioteca *Stable Baselines* para implementar o algoritmo de aprendizado por reforço. Essa biblioteca fornece uma série de algoritmos bastante utilizados, como DQN, PPO, A2C, entre outros. Você deve escolher um desses algoritmos e implementá-lo para resolver o ambiente "Mountain Car".

3. **Treinamento:** uma vez escolhido o algoritmo, o agente deve ser treinado utilizando o algoritmo escolhido para solucionar o ambiente. O treinamento deve ser executado por um número suficiente de episódios para permitir que o agente aprenda a superar os desafios do ambiente. Sugestão de treinamento com número de iterações 50K, 100K, 200K, 400K;

OPCIONAL: fique a vontade para treinar também um número maior de vezes tal como 1M a 2M (obs: com quantidades muito altas, o treino pode demorar um pouco mais, mas pode ter resultados melhores).

Relatório

Para **ambos os problemas 1 e 2**, seguem abaixo as especificações de como deve ser feita a análise dos resultados, o relatório e a entrega.

4. Análise de Resultados:

- **Obrigatório:** [80%] deve ser informada a configuração dos parâmetros utilizados para o treinamento, por exemplo: qual política utilizada e a quantidade necessária de episódios de treinamento para o agente poder resolver o problema, dentre outros. Além disso, deve ser apresentado um vídeo com o agente solucionando o problema no ambiente.
- **Desejável:** [100%] Monitoramento do desempenho do agente ao longo do tempo: um gráfico com as curvas de aprendizado ao longo dos episódios.
- **Se der tempo:** [110%] Uma análise mais minuciosa dos resultados pode incluir eventualmente a informação de quais algoritmos não funcionaram e sua justificativa e a comparação de diferentes hiperparâmetros ou algoritmos, entre outras métricas relevantes.

5. Relatório: elaborar um relatório descrevendo o processo de implementação e análise de cada problema. O relatório deve conter uma introdução, uma breve descrição dos ambientes “**Mountain Car**” e “**Car Racing**” e, para cada ambiente, a explicação do algoritmo escolhido, os detalhes da implementação, a análise dos resultados obtidos e as conclusões.

6. Entrega: o trabalho deve ser entregue na forma de um arquivo PDF contendo para cada problema:

- (a) relatório descrevendo o processo de implementação e análise dos problemas com os
- (b) códigos Python anexos ao fim do relatório e os
- (c) Links para os Notebooks Colab com os códigos-fonte da implementação funcionando, e com os
- (d) vídeos de testes para ambos; mas no caso do **Car Racing**, vídeo teste para as diferentes quantidades de iterações de treino.

Recomendações Gerais

- Os alunos podem consultar a documentação oficial da biblioteca GYMNASIUM e Stable Baselines para obter informações adicionais sobre o ambiente e os algoritmos disponíveis.
- Os alunos devem explorar diferentes configurações de hiperparâmetros para obter melhores resultados.
- É encorajado o uso de técnicas de pré-processamento de observações ou ações, se necessário.

Referências

Teoria

- <https://www.deeplearningbook.com.br/algoritmo-de-agente-baseado-em-ia-com-reinforcement-learning-parte-2/>

Ambientes:

- https://gymnasium.farama.org/environments/classic_control/mountain_car/
- https://gymnasium.farama.org/environments/box2d/car_racing/

SB3:

- <https://github.com/Stable-Baselines-Team/rl-colab-notebooks/tree/sb3>
- https://github.com/Stable-Baselines-Team/rl-colab-notebooks/blob/sb3/dqn_sb3.ipynb
- <https://stable-baselines.readthedocs.io/en/master/modules/dqn.html>
- <https://stable-baselines.readthedocs.io/en/master/modules/trpo.html>