Minigrid – Door Key – Uma análise de modelos (DQN x PPO)

Giancarlo Vanoni Ruggiero, Luciano Felix Dias, Tales Ivalque

Engenharia da Computação, INSPER

Prof. Fabrício Barth

Abstract—Neste projeto foi desenvolvido e validado um agente capaz de atuar no ambiente Door Key. O método de treinamento empregado foi ... Os resultados obtidos foram ...

keywords—*Reinforcement Learning, DQN, PPO, Minigrid, Door Key*

1. Introdução

O projeto tem como objetivo comparar o desempenho do ambiente single-agent Minigrid – Doorkey utilizando os algoritmos DQN (*Deep Q-Network*) e PPO (*Proximal Policy Optimization*). Neste ambiente, há uma chave que o agente deve adquirir para destrancar uma porta e chegar ao objetivo no quadrado verde.

2. Ambiente

Table 1. Direções do espaço de observação

ID	Vetor direção
0	$(1, 0)$
1	$(0, 1)$
2	$(-1, 0)$
3	$(0, -1)$

O espaço de observação é descrito por um dicionário contendo a direção do agente *direciton*, a imagem observável *image* e o espaço de missão *mission*. Sendo: *direction* um atributo discreto de valor como descrito na tabela 1; *image* sendo uma matriz RGB quadrada de valores inteiros contidos no intervalo $[0, 255]$ do tamanho do campo de visão do agente, e sendo este parametrizável como numero inteiro ímpar maior que 2 por padrão 7; e *mission* sendo o universo de ação do agente como uma matriz retangular de tamanho arbitrário contendo trincas seguindo a seguinte estrutura, respectivamente:

Table 2. Objetos do espaço de observação

ID	Nome	Objeto
0	*UNSEEN*	Não visível
1	*EMPTY*	Vazio
2	*WALL*	Parede
3	*FLOOR*	Chão
4	*DOOR*	Porta
5	*KEY*	Chave
6	*BALL*	Bola
7	*BOX*	Caixa
8	*GOAL*	Objetivo
9	*LAVA*	Lava
10	*AGENT*	Agente

- **ID do objeto** contido no latrilho atual, descrito como na tabela 2;
- **ID da cor** do latrilho atual, descrito como na tabela 3;
- **ID do estado** do objeto condito no latrilho atual, descrito como na tabela 4.

O espaço de ação é discreto com ações de movimento e poucas interações com elementos do ambiente. O espaço de ação é implementado conforme especificado na tabela 5.

Table 3. Cores do espaço de observação

ID	Nome	Vetor RGB
0	*RED*	$(255, 0, 0)$
1	*GREEN*	$(0, 255, 0)$
2	*BLUE*	$(0, 0, 255)$
3	*PURPLE*	$(112, 39, 195)$
4	*YELLOW*	$(255, 255, 0)$
5	*GREY*	$(100, 100, 100)$

Table 4. Estados do espaço de observação

ID	Nome	Estado
0	*OPEN*	Aberto
1	*CLOSED*	Fechado
2	*LOCKED*	Trancado

A função de recompensa é definida no domínio contínuo do intervalo $[0, 1]$ e proporcional a razão entre o número de passos e o número máximo de passos permitido para o agente atingir o estado final com sucesso, caso contrário a recompensa é anulada. Esta razão é definida com maior rigor na equação 1.

$$\text{RECOMPENSA} = \begin{cases} 1 - 0.9 \cdot \dfrac{\#\text{PASSOS}}{\max(\#\text{PASSOS})}, & \text{se sucesso} \\ 0, & \text{caso contrário} \end{cases} \tag{1}$$

3. Método

Para encontrar os resultados serão realizados treinamentos com diferentes números de episódios para cada algoritmo. Também será realizada a modificação do reward para que o agente consiga o reward não só chegando ao final mas realizando as ações necessárias para chegar lá, comparando os resultados.

Atenção

Para treinamento será utilizado a biblioteca *stable baselines*.

Também é importante destacar a implementação utilizada. Se foi uma implementação feita do zero pela equipe ou se foi utilizada uma biblioteca. Se a equipe optou por utilizar uma biblioteca é importante citar a biblioteca. Em ambos os casos, se a equipe fez a sua própria implementação ou se utilizou uma biblioteca, é importante **citar o repositório** onde os scripts se encontram.

Table 5. Espaço de ação

Número	Nome	Ação
0	*LEFT*	Vira para a esquerd
1	*RIGHT*	Vira para a direita
2	*FORWARD*	Move para a frente
3	*PICKUP*	Pega um objeto
4	*DROP*	Não utilizado
5	*TOGGLE*	Alternar/ativar um objeto
6	*DONE*	Não utilizado

Nesta seção também é importante descrever quais são os principais **indicadores** que a equipe está avaliando. Isto está diretamente relacionado com a função que pretende-se otimizar - que foi descrita na introdução deste relatório.

Eventualmente, a equipe deseja adicionar algum trecho de código nesta parte do relatório. Isto pode ser feito da seguinte maneira:

```
env = gym.make(
        'MountainCarContinuous-v0',
        render_mode='human')

(obs,_) = env.reset()

for i in range(1000):
    a, _s = model.predict(obs, deterministic=True)
    obs, reward, done, truncated, info = env.step(a)
    env.render()
    if done:
      obs = env.reset()
```

Code 1. Exemplo de código em Python

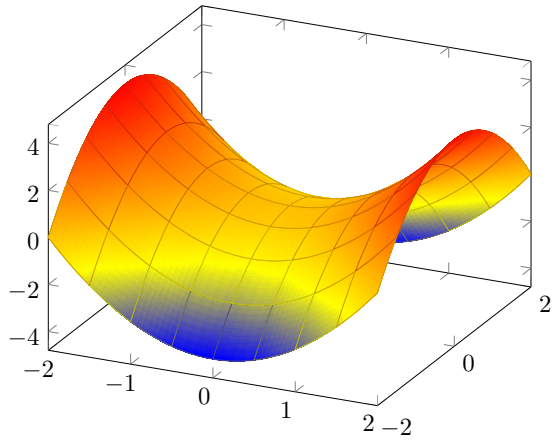

Figure 1. Exemplo de figura obtido a partir de *PGFPlots - A LaTeX package to create plots.* [Online]. Available: https://pgfplots.sourceforge.net/).

4. Resultados

O objetido desta pararte do relatório é descrever os resultados obtidos. Durante a disciplina nós vimos as principais técnicas para mostrar o aprendizado do agente e o desempenho do mesmo para realizar a tarefa. Esta seção precisa mostrar dados quantitativos que descrevem isto: a curva de aprendizagem e as validações feitas depois do treinamento.

Para apresentar os resultados obtidos com certeza a equipe terá que fazer uso de figuras, como a apresenta em 1.

Além de tabelas, como a apresentada em 6.

Table 6. Exemplo de tabela

Column 1	Column 2
Data 1	Data 2
Data 3	Data 4

5. Considerações finais

O objetivo deste trabalho foi ...