Projeto Final - Reinforcement Learning

Impactos do Curriculum Learning

Gabriel Valentim (xvalentim)

Maio 2025

1 Contexto e objetivo

O *Aprendizado por Reforço* (RL) é eficaz para treinar agentes em tarefas complexas, mas enfrenta dificuldades quando aplicado diretamente a ambientes desafiadores, devido à necessidade de aprender simultaneamente habilidades básicas. Para solucionar esse problema, este trabalho investiga o uso de *Curriculum Learning*, técnica que organiza o treinamento em tarefas progressivamente mais complexas, com o intuito de acelerar o aprendizado e melhorar a generalização. Avaliamos seu impacto em dois ambientes do *Gymnasium* — *LunarLander-v3* e *BipedalWalker-v3* — comparando agentes treinados com e sem (*baseline*) essa abordagem, a fim de verificar ganhos em eficiência, convergência e desempenho das políticas aprendidas.

2 Metodologia

O uso de *curriculum* nas aplicações tem um conceito bem definido, porém é personalizado no contexto de cada ambiente analisado. Isto é, a depender do tipo de tarefa a ser realizada, a divisão de tarefas menores e menos complexas precisa ser arquitetada de acordo com contexto.

Nesse sentido, o procedimento para o trabalho presente será baseado em analisar o contexto e dividir tarefas manualmente que fazem sentido para aquele ambiente, de acordo com o diagrama 1.

Por fim, a avaliação expressa em *evaluate* no diagrama da figura 1 é feita comparando os gráficos de recompensa ao longo do treinamento, bem como o gráfico os gráficos de entropia inserida na *loss*. Para critérios de comparação, vemos os dois pontos abaixo:

- **Tempo de convergência**
- **Valor final de recompensa média**

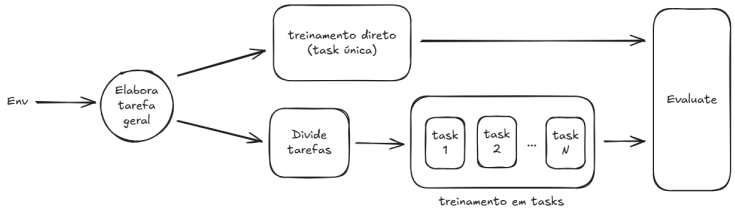


Figure 1: Fluxograma de procedimento aplicado para avaliar o impacto do *curriculum* na tarefa.

3 Ambientes utilizados

3.1 Lunar Lander

O ambiente *LunarLander-v3* simula a tarefa de pouso controlado de um módulo espacial em uma superfície plana com um ponto de pouso centralizado. O agente controla três motores (um principal e dois laterais) e deve aprender a pousar suavemente, com mínima inclinação e velocidade, enquanto economiza combustível. A cada episódio, o lander inicia em uma posição aleatória com velocidade e ângulo variáveis. O espaço de observação é composto por 8 variáveis contínuas que incluem posição, velocidade, ângulo, velocidade angular e contatos das pernas com o solo. O espaço de ação pode ser discreto (quatro ações: sem ação, motor esquerdo, motor principal, motor direito) ou contínuo, dependendo da configuração. O episódio termina quando o agente pousa com sucesso, colide com o solo ou sai da área de pouso.

3.2 Bipedal Walker

O ambiente *BipedalWalker-v3* simula um robô bípede que deve aprender a caminhar sobre um terreno irregular. O agente possui quatro juntas motoras (duas nos quadris e duas nos joelhos), e seu objetivo é se deslocar o mais longe possível sem cair. O espaço de observação é composto por 24 variáveis contínuas que incluem ângulo do tronco, velocidades lineares e angulares, ângulos e velocidades das articulações, contatos das pernas com o solo, além de sensores LIDAR que mapeiam a proximidade do solo à frente. As ações são valores contínuos no intervalo $[-1, 1]$ que controlam as velocidades-alvo das juntas. Há também uma versão mais difícil, chamada *hardcore*, que inclui obstáculos como escadas, buracos e plataformas elevadas, tornando o aprendizado mais desafiador.

4 Distribuição de tarefas

4.1 Lunar Lander

Para o *lunar lander*, foram estipuladas 4 tarefas, que modificam alguns parâmetros do ambiente de tal forma que podemos organizar sua ordem por nível de difi-

culdade. Os parâmetros que modificamos foram: **gravidade**, **força do vento** e **força da turbulência**. Com a lista de dicionários, é possível identificar a quebra.

```
lunarlander_stages = [
    {
        "name": "easy",
        "gravity": -5.0,
        "enable_wind": False,
        "wind_power": 0.0,
        "turbulence_power": 0.0,
        "parcial_timesteps": 100_000,
    },
    {
        "name": "intermediate",
        "gravity": -10.0,
        "enable_wind": False,
        "wind_power": 0.0,
        "turbulence_power": 0.0,
        "parcial_timesteps": 100_000,
    },
    {
        "name": "hard_with_wind",
        "gravity": -10.0,
        "enable_wind": True,
        "wind_power": 5.0,
        "turbulence_power": 0.5,
        "parcial_timesteps": 100_000,
    },
    {
        "name": "full_difficulty",
        "gravity": -10.0,
        "enable_wind": True,
        "wind_power": 15.0,
        "turbulence_power": 1.5,
        "parcial_timesteps": 100_000,
    }
]
```

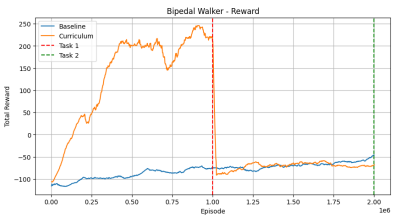
4.2 Bipedal Walker

Para o ambiente presente, foram criadas apenas duas tarefas. Uma que conta com o ambiente do *bipedal walker* apenas como uma trajetória com obstáculos e resistências e outra que não consta.

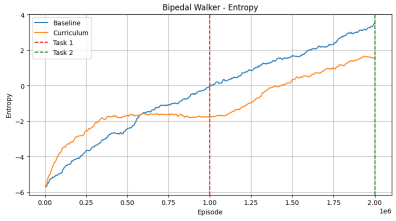
```
bipedalwalker_stages = [
    {
        "name": "easy",
        "hardcore": False,
        "parcial_timesteps": 250_000,
    },
    {
        "name": "hardcore",
        "hardcore": True,
```

```
        "parcial_timesteps": 250_000,
    },
]
```

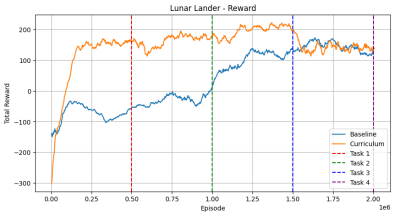
5 Resultados



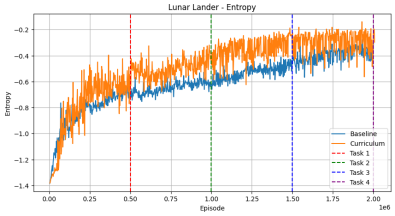
(a) Recompensa Acumulada



(b) Entropia



(c) Recompensa Acumulada



(d) Entropia

Figure 2: Comparativo de treinamento *baseline* e com *curriculum*.

Dado o contexto introduzido pelas seções anteriores, o esperado era que, após divisão de tarefas, o agente iria convergir de maneira mais rápida ou até mesmo em patamares superiores quando comparado com um processo dito como "*baseline*", isto é, treinado em uma tarefa só. Porém, isso não foi observado em nossos experimentos.

Acompanhando a figura 2 é possível observar alguns resultados do ponto de vista de recompensa acumulada e entropia. Note que, apesar da recompensa crescer rapidamente para o caso de uso de *curriculum*, o patamar no qual a tarafe mais complexa se aproxima é parecido com o nível de recompensa atingido pelo método *baseline*. Mostrando que, para os experimentos presentes (e ambientes testados), não houve diferença significativa na recompensa acumulada.

Do ponto de vista de entropia, é possível notar uma diferença para o caso do *BipedalWalker*. É possível que a divisão entre as duas tarefas (terreno liso e terreno com resistência) possa ter diminuído sensivelmente a entropia da tomada de decisão. Isso pode significar que a tomada de decisão da *policy* se tornou menos incerta, o que é bom no sentido de ser um indício que a policy aprendeu a tarefa anterior, porém pode prejudicar o agente no sentido de explorar novas ações.