Instituto Tecnológico de Aeronáutica - ITA
CT-213 - Inteligência Artificial aplicada à Robótica Móvel
Aluno: Ulisses Lopes da Silva

Relatório do Laboratório 12 - Deep Q-Learning

---

1 Breve Explicação em Alto Nível da Implementação

A implementação proposta visa resolver o ambiente *MountainCar-v0* do Gym utilizando o algoritmo **Deep Q-Learning (DQN)**, uma das abordagens fundamentais do Aprendizado por Reforço Profundo. Nesse problema, o agente deve aprender uma política que maximize a recompensa acumulada ao empurrar um carro até o topo de uma colina, usando uma rede neural para estimar os valores de ação $Q(s, a)$.

A classe central, `DQNAgent`, concentra a lógica do agente e incorpora os componentes essenciais do DQN. O método `make_model()` implementa a rede neural com duas camadas ocultas densas de 24 neurônios cada, utilizando a ativação ReLU, seguidas por uma camada de saída com ativação linear, compatível com o número de ações possíveis. A rede é compilada com o otimizador Adam e função de perda MSE, conforme parâmetros estabelecidos no roteiro do laboratório.

O método `act()` implementa a política ε-*greedy*, permitindo que o agente explore com probabilidade ε ou escolha a ação de maior valor Q estimado com $1 - \varepsilon$. Para evitar poluição no terminal durante a execução, o parâmetro `verbose=0` foi utilizado nas predições do modelo.

Durante o treinamento, o agente armazena experiências na memória de repetição (*replay buffer*) por meio do método `append_experience()`. O método `replay()` realiza o aprendizado propriamente dito: amostras são retiradas da memória, os alvos de aprendizado são calculados com base na recompensa e no valor máximo estimado para o próximo estado, e a rede neural é ajustada via *backpropagation*.

O arquivo `utils.py` define a função `reward_engineering_mountain_car`, que modifica a recompensa original do ambiente. São adicionados termos proporcionais à distância ao quadrado desde a posição inicial e ao quadrado da velocidade, incentivando o movimento e o acúmulo de momento. Um bônus adicional é dado quando o objetivo (posição ≥ 0.5) é atingido.

De forma geral, a implementação obedece às especificações do roteiro do laboratório, incluindo *reward shaping* e seleção de ações com ε-*greedy*.

2 Figuras Comprovando Funcionamento do Código

2.1 Sumário do Modelo

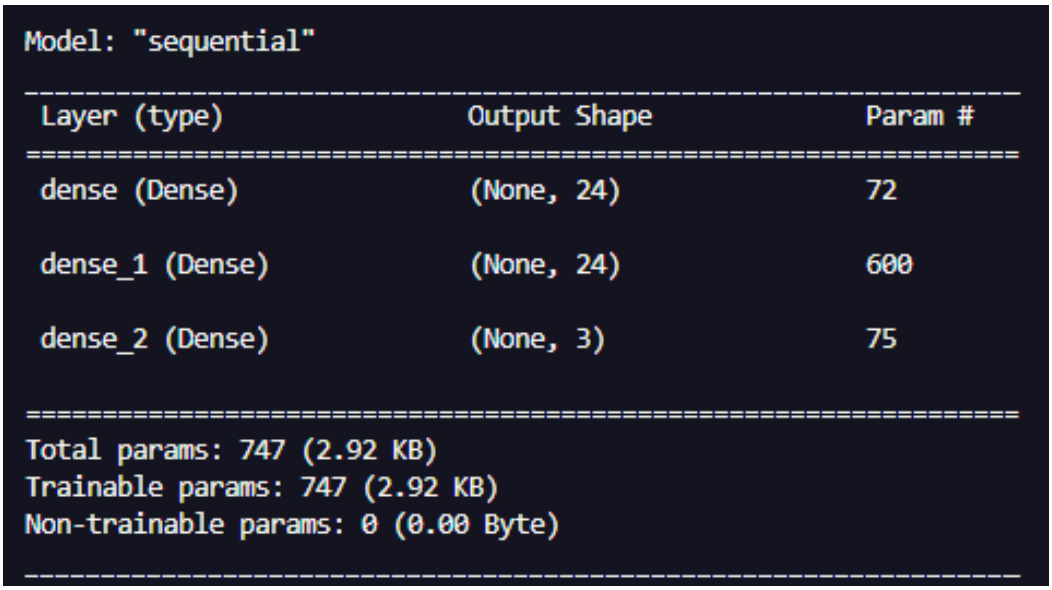

Fig. 1: Sumário do modelo obtido na saída do terminal

Tabela 1: Sumário da Arquitetura do Modelo de Rede Neural

Camada (tipo)	**Forma de Saída**	**Nº de Parâmetros**
`dense (Dense)`	(None, 24)	72
`dense_1 (Dense)`	(None, 24)	600
`dense_2 (Dense)`	(None, 3)	75
Total de parâmetros		**747 (2.92 KB)**
Parâmetros treináveis		**747 (2.92 KB)**
Parâmetros não-treináveis		**0 (0.00 Byte)**

2.2 Retorno ao Longo dos Episódios de Treinamento

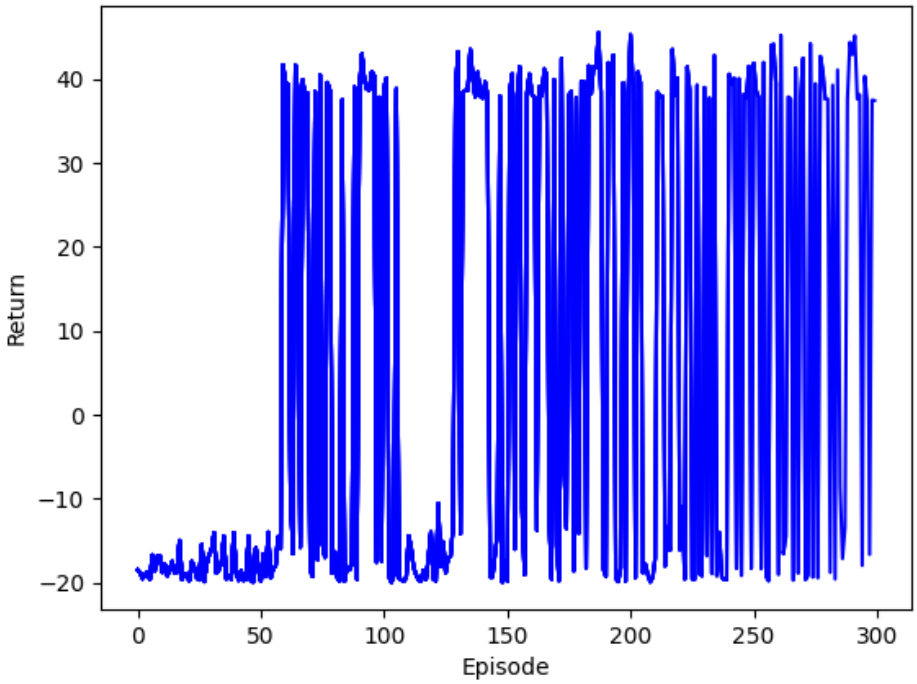

Fig. 2: Evolução do retorno ao longo dos episódios de treinamento com DQN. Observa-se melhora significativa do desempenho por volta do episódio 60 e estabilização posterior com oscilações.

2.3 Política Aprendida pelo DQN

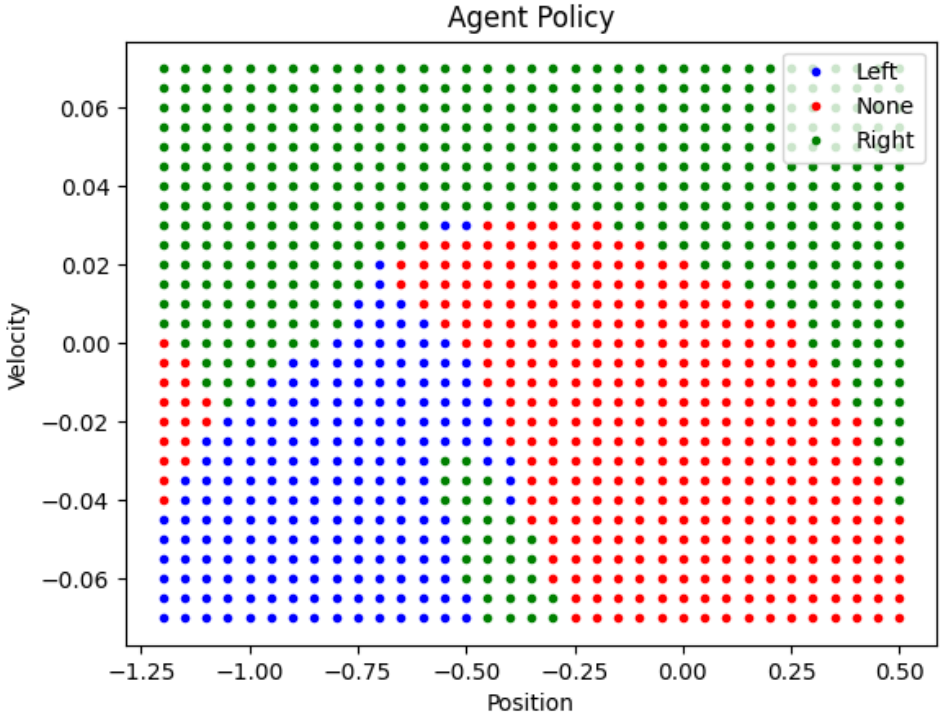

Fig. 3: Política gulosa aprendida ao final do treinamento. Ações de empurrar para a esquerda (azul), direita (vermelho) ou inércia (verde) são bem distribuídas conforme o espaço de estados.

2.4 Retorno de 30 Episódios Usando a Rede Neural Treinada

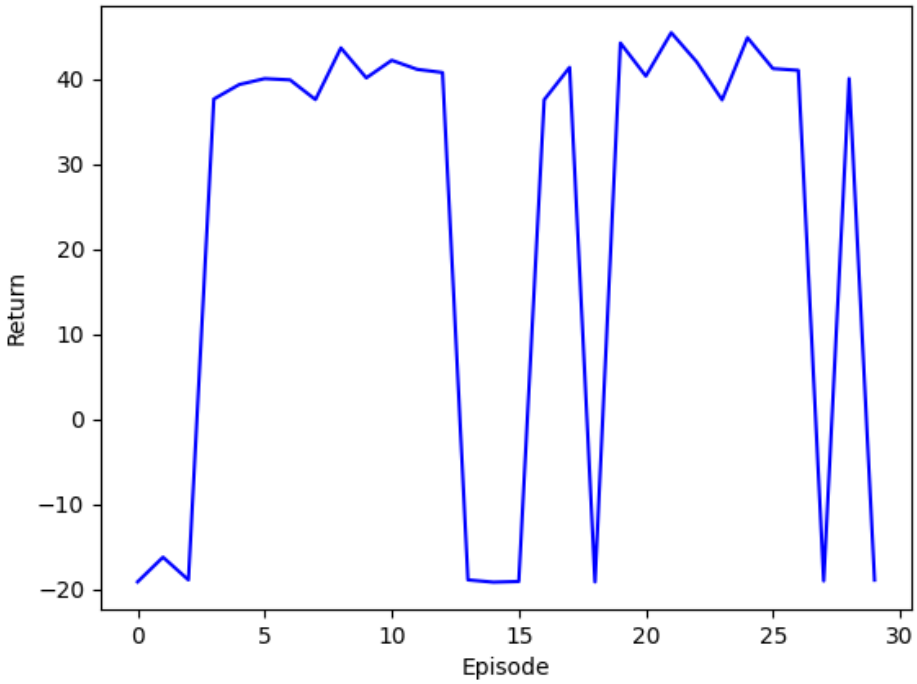

Fig. 4: Retorno obtido durante os episódios de validação com política determinística. Os altos retornos indicam que a política aprendida é eficiente na resolução do ambiente.

3 Discussão dos Resultados

As figuras obtidas a partir dos experimentos com o algoritmo Deep Q-Networks (DQN) no ambiente `Mountain Car` evidenciam o processo de aprendizagem do agente, assim como a política aprendida e o desempenho nos episódios de validação.

Na Fig. 2, observa-se a evolução do retorno ao longo de 300 episódios de treinamento. Inicialmente, os retornos acumulados são baixos, próximos de -20, refletindo a dificuldade do agente em alcançar a bandeira no topo da montanha. Contudo, por volta do episódio 60, há uma oscilação significativa no retorno, e a partir do episódio 100 os retornos positivos tornam-se mais frequentes. Esse comportamento vai ao encontro do esperado: durante as primeiras interações, o agente explora o ambiente com uma política altamente estocástica (alto ϵ), e com o tempo, à medida que o ϵ diminui, a política torna-se mais determinística, favorecendo a exploração das trajetórias que conduzem ao sucesso. A alternância entre altos e baixos retornos nas fases finais pode estar associada à aleatoriedade presente na política.

A Fig. 3 apresenta a política gulosa aprendida pelo agente ao final do treinamento. Ela mostra a ação escolhida (empurrar para a esquerda, não fazer nada, ou empurrar para a direita) para cada combinação de posição e velocidade. A política segue um padrão esperado em que,

nas regiões mais à esquerda do espaço de estados, o agente tende a empurrar o carro para a esquerda (em azul), a fim de ganhar momento suficiente para escalar o lado direito. Conforme o carro avança para a direita, a política muda gradualmente, favorecendo ações de empurrar para a direita (vermelho). A ação de inércia (verde) é escolhida principalmente em estados centrais e de baixa velocidade, o que é coerente com uma política que busca otimizar o acúmulo de momento.

Por fim, a Fig. 4 mostra o desempenho do agente durante 30 episódios de avaliação, utilizando a política puramente gulosa ($\epsilon = 0$). Nota-se que, na maioria dos episódios, o agente obtém retornos próximos de 40, indicando que ele é capaz de resolver o ambiente de maneira consistente. Oscilações ocasionais nos episódios de avaliação indicam que, embora a política seja eficaz, ainda existem estados iniciais em que o agente pode ter dificuldades, o que é comum no ambiente Mountain Car devido à sua natureza determinística sensível às condições iniciais.

Em conjunto, os resultadoscomprovam a eficácia da implementação do DQN, demostrando aprendizado progressivo, construção de uma política coerente e desempenho razoavelmente estável em avaliação, que condiz com algoritmos como os que foram estudados em aula.