Aprendizado por Reforço
Implementação

Valdinei Freire

(EACH - USP)

GYM AI

1. env = gym.make('CartPole-v0')
2. observation = env.reset()
3. while condição
   (a) escolhe action
   (b) observation, reward, done = env.step(action)

Ambiente Discreto

observation, reward, done = env.step(action)

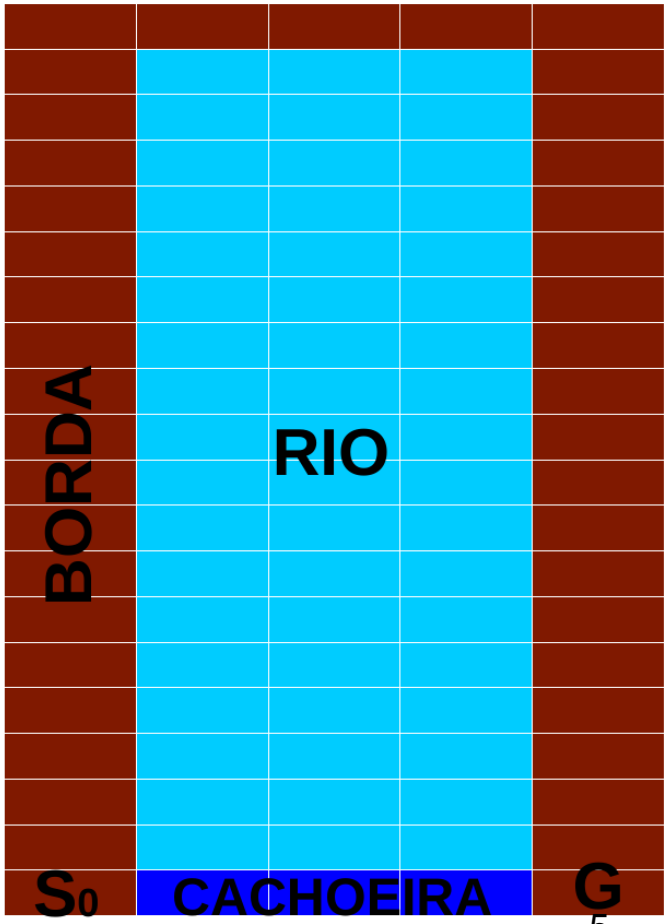
1. x $\leftarrow$ rand()
2. sum $\leftarrow$ 0
3. sNext $\leftarrow$ 0
4. while sum $<$ x
   (a) sNext $\leftarrow$ sNext+1
   (b) sum $\leftarrow$ sum + T(env.s,action,sNext)
5. observation $\leftarrow$ observe(sNext)
6. reward $\leftarrow$ R(env.s,action,sNext)
7. env.s $\leftarrow$ sNext
8. if sNext $\in \mathcal{G}$
   (a) done $\leftarrow$ true
9. else
   (a) done $\leftarrow$ false

Q-Learning com ϵ-greedy

1. Inicialize $Q(s, a)$ arbitrariamente
2. env = gym.make('FrozenLake8x8-v0')
3. while condição
   (a) s = env.reset()
   (b) while $d \neq true$
      i. if $rand > \epsilon$
         A. $a \leftarrow \arg\max_{a \in \mathcal{A}} Q(s, a)$
      ii. else
         A. $a \leftarrow rand$
      iii. s, r, d = env.step(action)
      iv. $\delta \leftarrow r + \gamma \max_{a' \in \mathcal{A}} Q(s, a') - Q(s, a)$
      v. $Q(s, a) \leftarrow Q(s, a) + \alpha\delta$

Ambiente - TRAVESSIA DO RIO

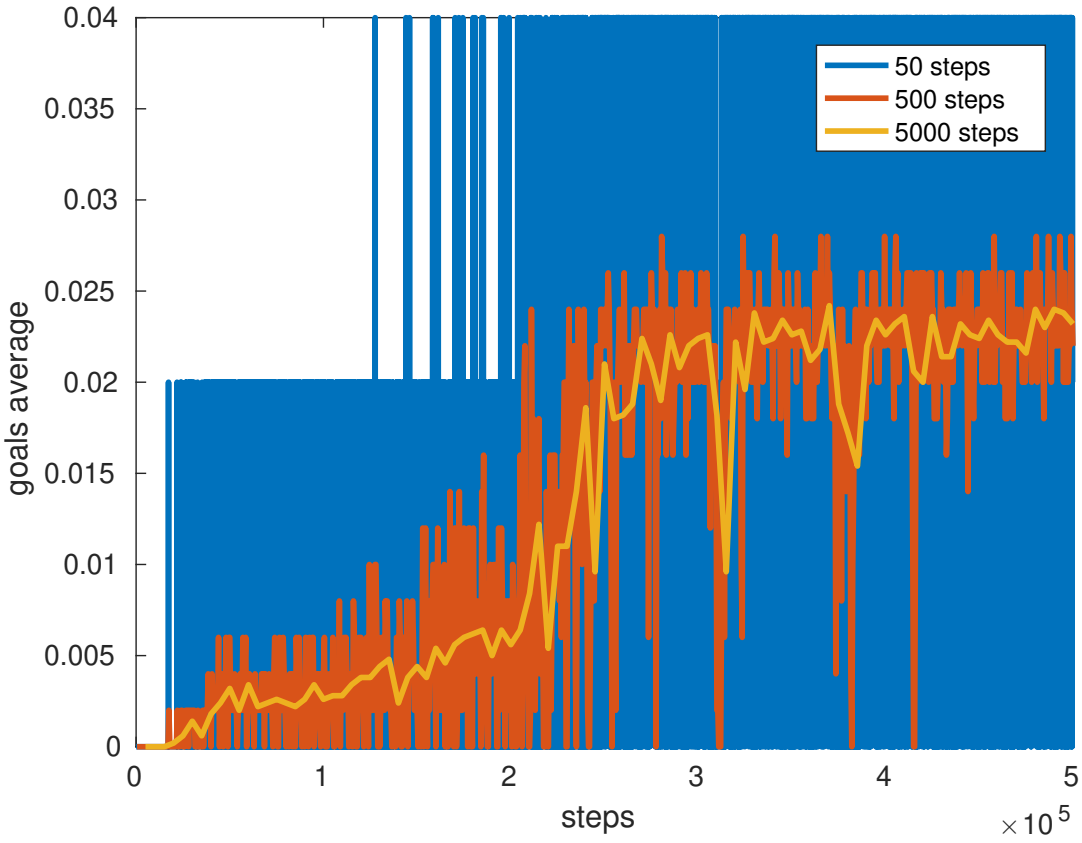
- Borda: resultados das ações são deterministas
- Rio: por causa da correnteza sempre há uma chance de ir na direção da cachoeira.
- Cachoeira: volta para o estado inicial
- Instância: 20x100



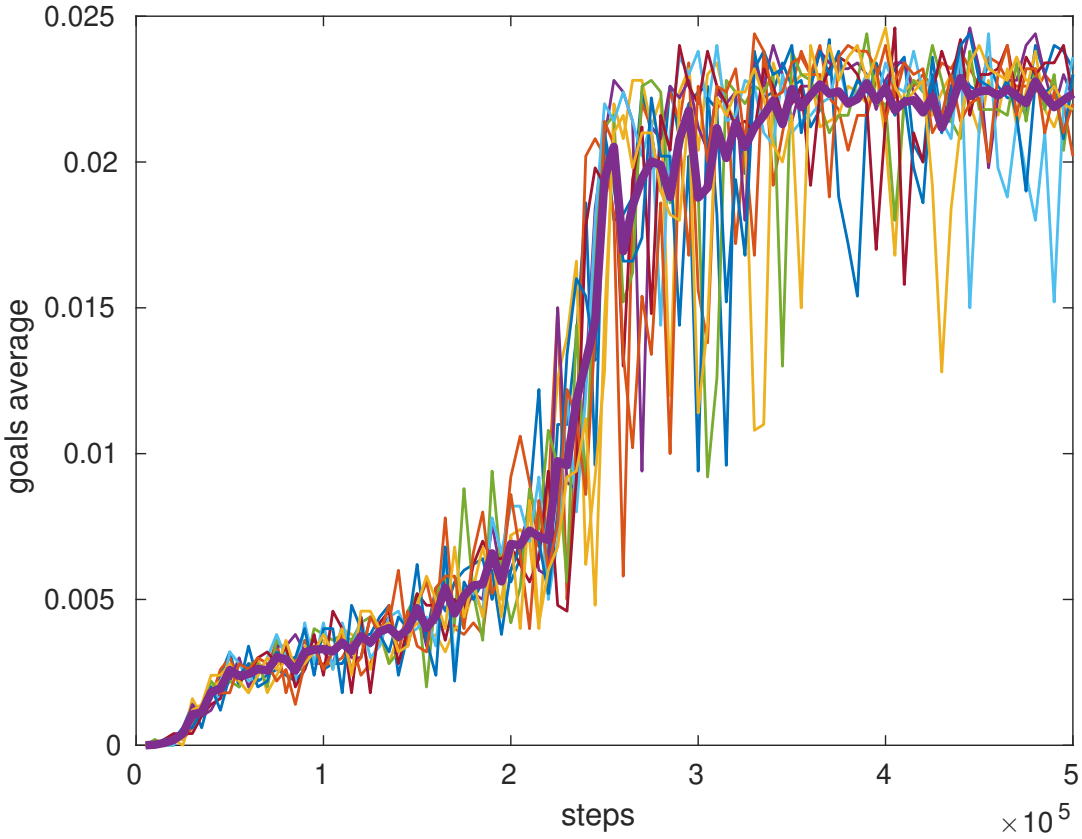
Exibição de Resultados

- Média de várias execuções
- recompensa média por passos
- passos para chegar na meta
- recompensa acumulada

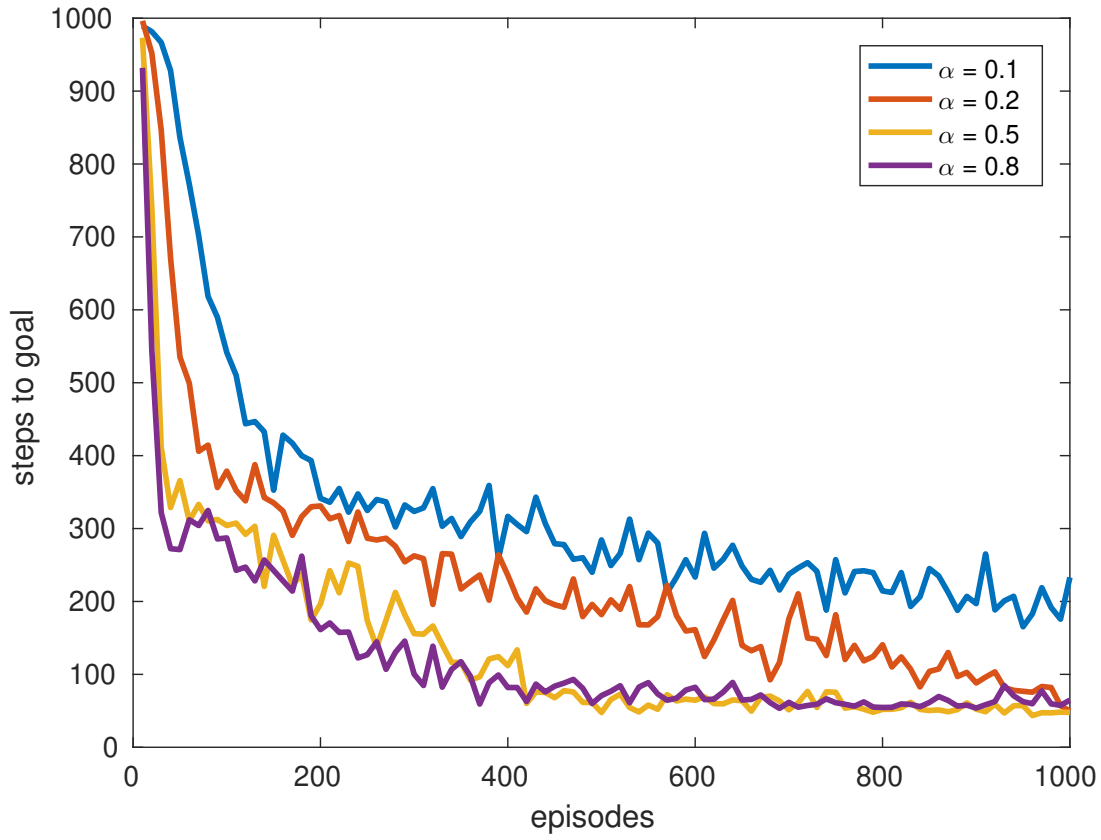
Q-Learning: média de recompensa e resolução do gráfico



Q-Learning: 10 execuções



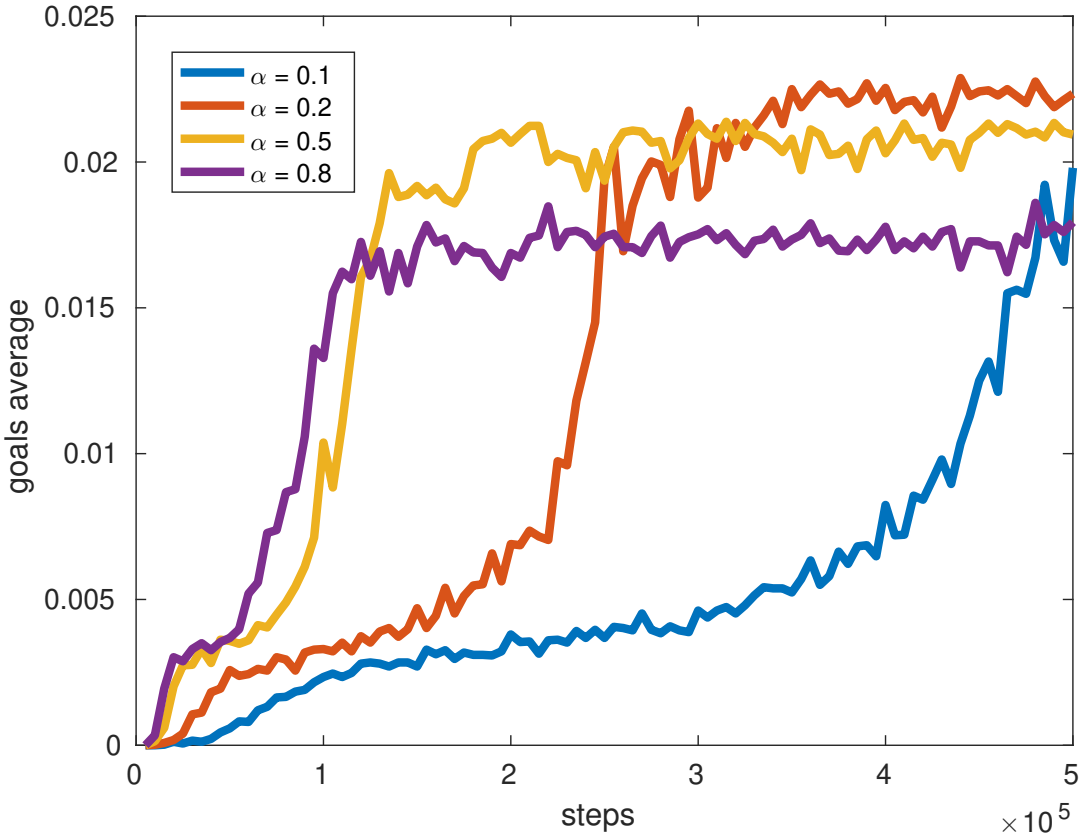
Q-Learning: média de passos para a meta



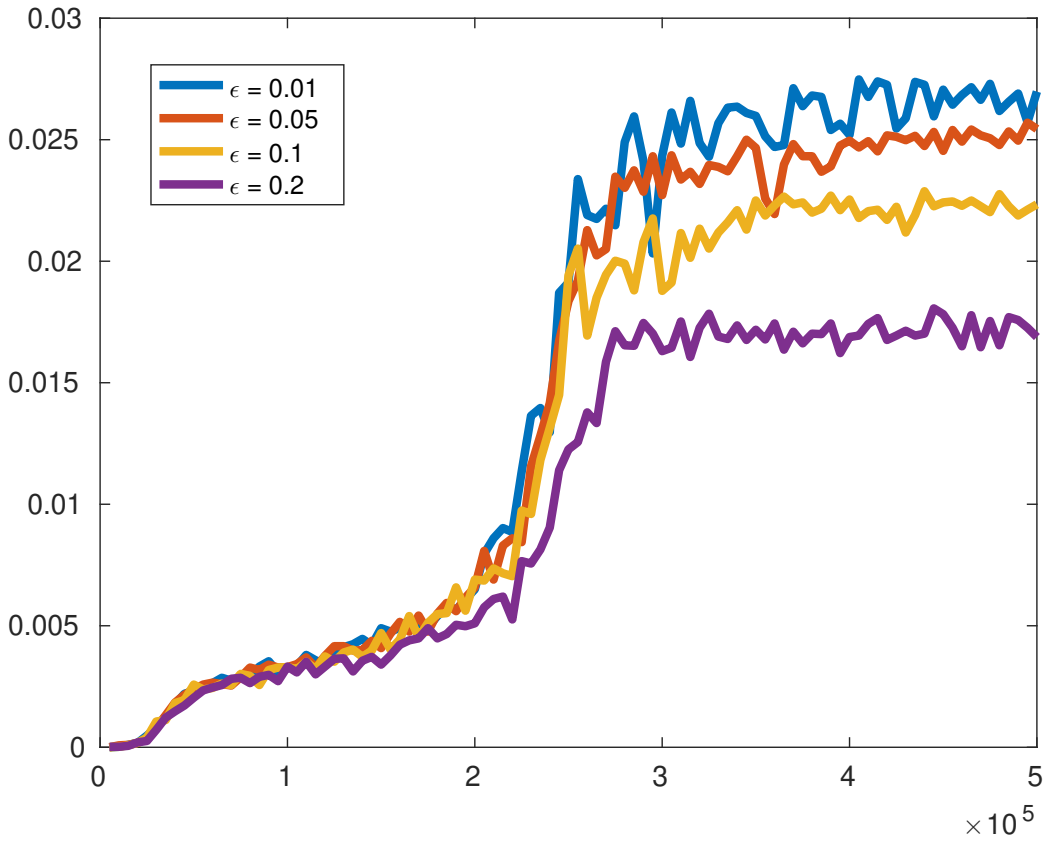
Q-Learning: parâmetros

- taxa de aprendizado: α
- taxa de exploração: ϵ
- objetivo: recompensa vs custo
- valor inicial de Q: mínimo ou máximo

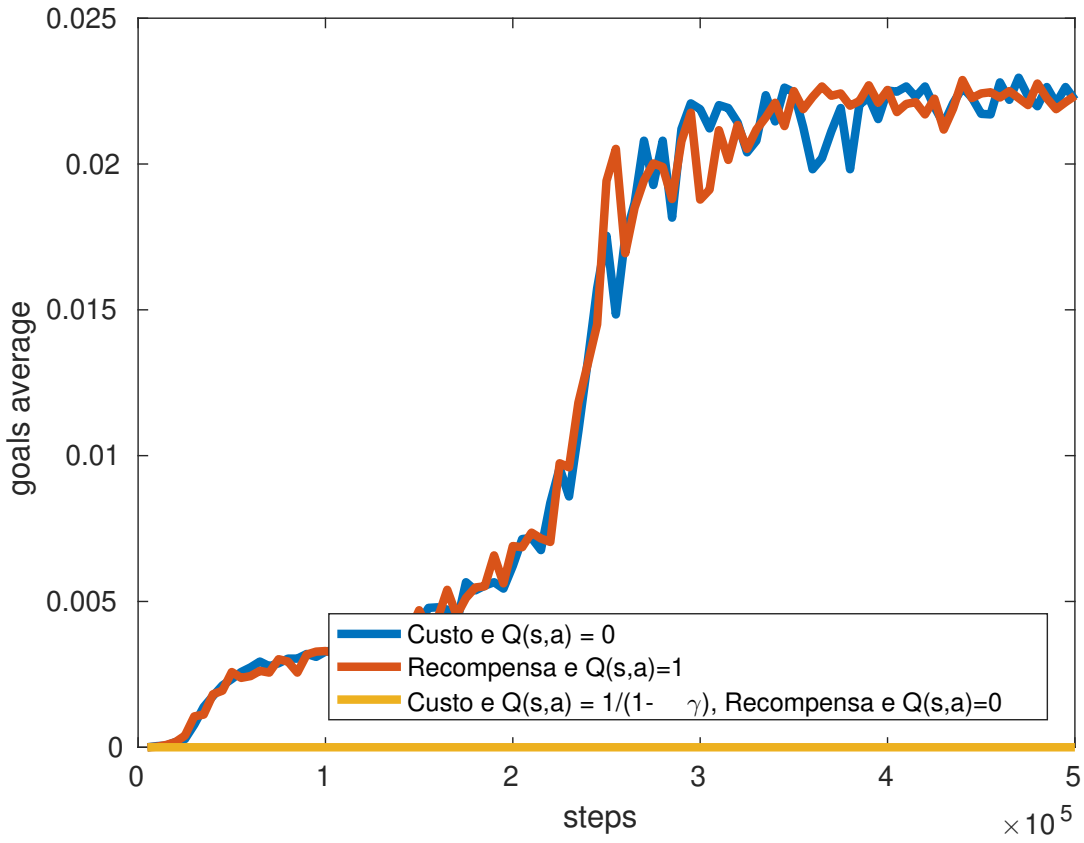
Q-Learning: taxa de aprendizado



Q-Learning: taxa de exploração



Q-Learning: inicialização



SARSA(λ)