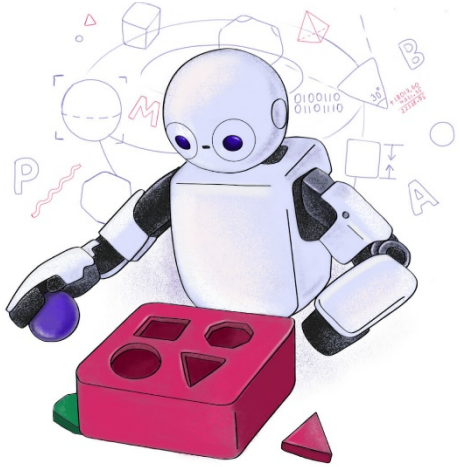
TP558 - Tópicos avançados em Machine Learning:

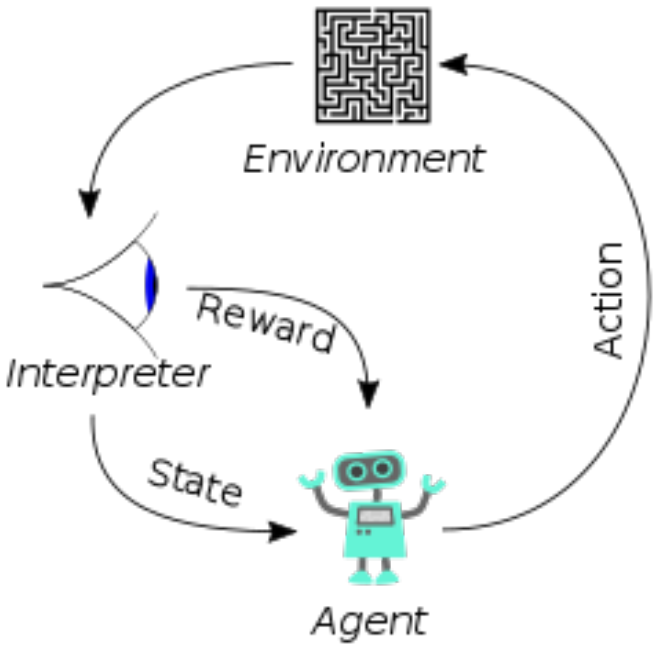
Ator-Crítico



Inatel

Pedro Marcio Raposo Pereira
pedro.marcio@inatel.br

O que é aprendizado por reforço?



https://en.wikipedia.org/wiki/Reinforcement_learning

Aprendizado por reforço (*Reinforcement learning* - RL) é o estudo dos agentes e como eles aprendem por tentativa e erro.

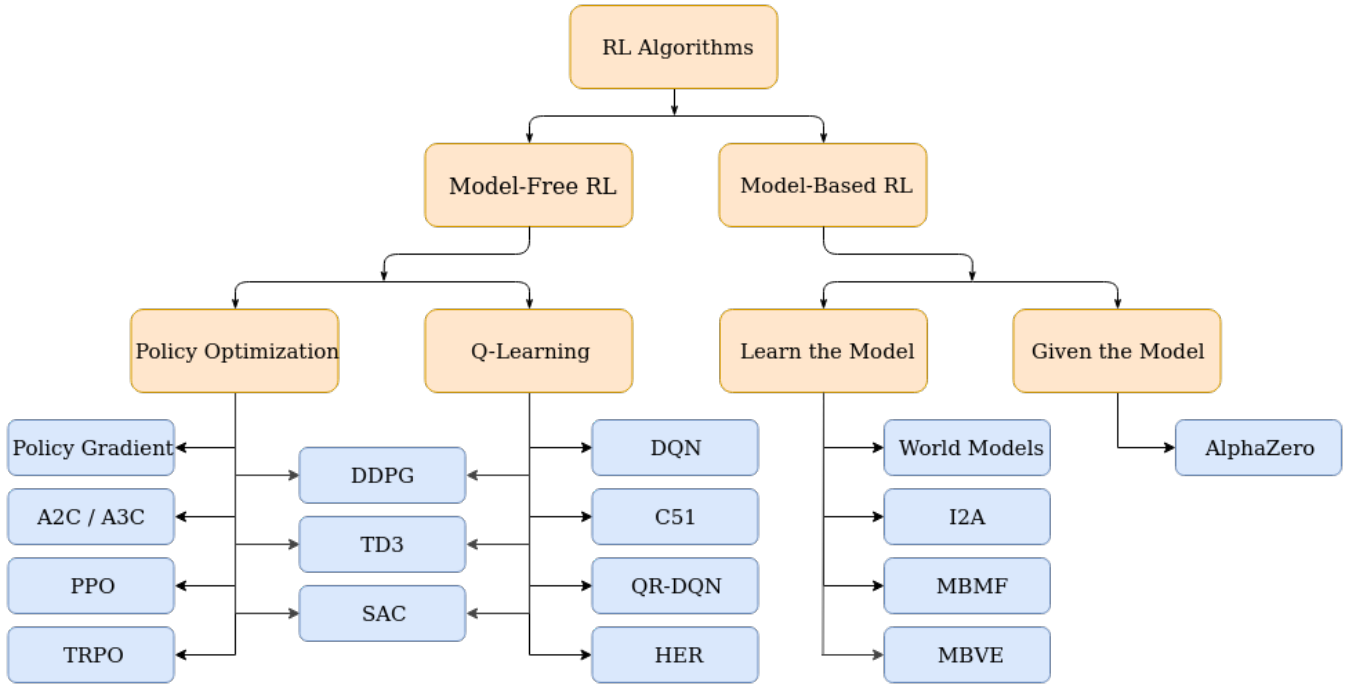
Premiar ou punir um agente pelos seu comportamento faz com que seja mais provável que ele repita ou evite o mesmo comportamento no futuro.

Principais conceitos e terminologias

- Os personagens principais do RL são o agente (*agent*) e o ambiente (*environment*). O ambiente é o mundo no qual o agente vive e interage. A cada passo de interação, o agente vê uma observação de um estado (*state*) e então decide qual ação (*action*) tomar. O ambiente pode mudar por conta própria ou reagir a uma ação do agente.
- O agente também recebe uma recompensa (*reward*) do ambiente, um número que quantifica quão boa ou ruim o estado atual é. O objetivo do agente é maximizar a premiação cumulativa, também chamada de retorno (*return*).

Principais conceitos e terminologias

- Métodos de RL são as maneiras que o agente pode aprender os comportamentos que o levam a esse objetivo.



TAXONOMIA DOS MODELOS DE RL: https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html

Métodos de Gradiente de Política

- Os métodos de gradiente de política são muito comuns em algoritmos de aprendizado por reforço onde o modelo do ambiente não é conhecido.
- Eles ajudam o agente a aprender qual a melhor ação tomar em uma determinada situação.
- Além disso, o método de gradiente de política desempenha o papel do "ator" em métodos chamados Ator-Crítico, onde o "crítico" avalia as ações tomadas pelo "ator".

Métodos de Gradiente de Política

- Basicamente, os métodos de gradiente de política atualizam a distribuição de probabilidade das ações para que as ações com maior recompensa esperada tenham um valor de probabilidade maior para um estado observado.
- Em outras palavras, eles ajustam as chances de tomar diferentes ações em uma situação específica, de modo que as ações mais promissoras sejam mais prováveis de serem escolhidas.

Métodos de Gradiente de Política

- A função objetivo para os gradientes de política é definida como:

$$\nabla_\theta J(\theta) = \underset{\tau}{E}\left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta\left(a_t \middle| s_t\right) G_t\right]$$

Em que π_θ é a política, a_t é a ação, s_t é o estado e G_t é o ganho acumulado, definido como

$$G_t = \sum_{t'=t+1}^{T} \gamma^{t'-t-1} r_{t'}$$

γ é o fator de desconto e r_t a recompensa [1][2].

Algoritmo *Reinforce*

- O algoritmo Reinforce é uma técnica usada para ensinar um agente a realizar ações em um ambiente para maximizar recompensas futuras.

Algoritmo 1: Método do gradiente de política de Monte-Carlo

Requer: política diferenciável $\pi(a|s,\theta)$

início
  Inicializar a política com parâmetros θ;
  enquanto *Verdadeiro* **faça**
    Gerar um episódio $s_0, a_0, r_1, \ldots, s_{T-1}, a_{T_1}, r_T$ de acordo com $\pi(\theta)$;
    para $t = 0, \ldots, T-1$ **faça**
      $\theta \leftarrow \theta + \alpha\gamma^t G \nabla_\theta \log(\pi(a_t|s_t,\theta))$;
    fim
  fim
fim

Algoritmo *Reinforce*

1 - Realize uma trajetória: O agente executa ações com base na sua política atual (ou seja, nas regras que ele segue) e observa o que acontece em cada etapa

2 - Guarde informações importantes: Em cada etapa da trajetória, o agente registra duas coisas: a probabilidade das ações que ele tomou e as recompensas que recebeu.

3 - Calcule as recompensas futuras: Agora, o agente olha para frente e calcula quanta recompensa ele espera receber no futuro. Isso inclui recompensas imediatas e também recompensas que virão depois.

4 - Atualize a política: Com base nas informações coletadas, o agente ajusta suas regras (ou política) para tornar mais provável que ele tome ações que levem a recompensas maiores.

5 - Repita o processo: O agente continua repetindo esses passos, atualizando sua política à medida que aprende mais sobre o ambiente e busca maximizar suas recompensas.

Algoritmo *Reinforce*

No algoritmo Reinforce, usamos amostras aleatórias (Monte Carlo) para atualizar como nosso agente toma decisões. Isso significa que, durante o treinamento, o agente experimenta diferentes caminhos e toma decisões com base neles.

Aqui está a questão: como esses caminhos variam aleatoriamente, nossas estimativas sobre quão boas são as ações podem variar muito. Essa variabilidade faz com que nossos cálculos sobre quão boas ou ruins são as ações fiquem um pouco ruidosos (ou seja, imprecisos), o que pode tornar o aprendizado instável.

Outro problema surge quando o agente não recebe recompensas significativas. Se ele não está ganhando nada com suas ações, como pode saber se está agindo de forma correta ou não? Ele precisa de algum tipo de feedback para aprender, mas se suas ações não resultarem em recompensas, ele fica “perdido”.

Reduzindo a Variância

Uma maneira de reduzir a variância da estimação do gradiente é a introdução de uma função patamar (*baseline function*) $b(s_t)$ que deve ser subtraída do retorno acumulado, e portanto resultam em

$$\nabla_\theta J(\theta) = \underset{\tau}{E}\left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t)\left(G_t - b(s_t)\right)\right]$$

Intuitivamente, estamos fazendo os gradientes menores. Matematicamente, a inclusão da função patamar resulta em um estimador de menor variância e sem viés [3].

Método Ator-Crítico

Para entender como o método do ator-crítico funciona, vamos lembrar, que o gradiente é expresso por

$$\nabla_\theta J(\theta) = \underset{\tau}{E}\left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta\left(a_t \middle| s_t\right) G_t\left(a_t, s_t\right)\right]$$

Após, algumas manipulações matemáticas [1, 2] é possível escrever que

$$\nabla_\theta J(\theta) = \underset{\tau}{E}\left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta\left(a_t \middle| s_t\right) Q\left(a_t, s_t\right)\right]$$

em que $Q(a_t, s_t)$ é a função Q (Função de estado-ação).

Método Ator-Crítico

Sabemos que é matematicamente impossível encontrar uma expressão para determinar o valor de $Q(a_t, s_t)$, mas sabemos que as redes neurais são aproximadores universais de funções. Podemos parametrizar uma rede neural para estimar a função $Q_\phi(a_t, s_t)$ e portanto teremos que

$$\nabla_\theta J(\theta) = E_\tau \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t | s_t) Q_\phi(a_t, s_t) \right]$$

Método Ator-Crítico

Nos Métodos Ator-Crítico, temos dois papéis principais:

O "Crítico": Este é o responsável por avaliar as ações tomadas pelo agente. Ele estima o valor de diferentes ações ou estados.

O "Ator": Este é o responsável por decidir quais ações o agente deve tomar. Ele ajusta a política com base no feedback do Crítico.

Ambos o Crítico e o Ator usam redes neurais para fazer essas estimativas e decisões. Quando dizemos que é um "Ator-Crítico Q", significa que estamos usando uma rede neural para estimar os valores Q.

Método Ator-Crítico

Como vimos anteriormente, podemos utilizar de funções de patamar para reduzir a variância do estimador.

Se utilizarmos a função de valor V como patamar, teremos

$$\nabla_\theta J(\theta) = \underset{\tau}{E}\left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta\left(a_t \middle| s_t\right)\left(Q_\phi\left(a_t, s_t\right) - V\left(s_t\right)\right)\right]$$

Método Ator-Crítico

A função de vantagem $A^{\pi}(s,a)$correspondente à politica π descreve quão melhor é tomar uma ação específica em um estado, sobre uma ação selecionada aleatoriamente, assumindo que a política seja mantida para sempre. Matematicamente:

$$A^{\pi}(s,a)=Q^{\pi}(s,a)-V^{\pi}(s)$$

Por meio das equações de Bellman, ainda é possível escrever que

$$A^{\pi}(s,a)=r_{t+1}+\gamma V_{\phi}(s_{t+1})-V_{\phi}(s_t)$$

Note que é necessário apenas uma rede neural para estivar a vantagem.

Método Ator-Crítico

Dessa forma, podemos expressar a estimação dos gradientes como:

$$\nabla_\theta J(\theta) \sim \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t)\left(r_{t+1} + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)\right)$$
$$= \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) A(s_t, a_t)$$

Como estamos estimando os valores da função vantagem, podemos dizer que esse é um *Advantage Actor-Critic (A2C)*.

Método Ator-Crítico

O Erro TD (*Temporal Difference*) é expressado como

$$\delta_t = r_{t+1} + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$$

Essa é a mesma expressão que utilizamos para computar a vantagem anteriormente.

Como utilizamos o crítico para prever o valor do estado, podemos usar o próprio erro TD como função de custo. Como o objetivo é reduzir o erro, podemos fazer $J(\phi) = \delta^2$ e assim obter

$$\phi = \phi + \beta \delta \nabla_\phi V_\phi(s)$$

Alternativamente, pode-se usar a perda de Huber ao invés do erro quadrático, já que ela é menos sensitiva.

Método Ator-Crítico

Algoritmo 2: Ator-Crítico

Requer: política diferenciável $\pi(a|s,\theta)$, valor $V_\phi(s)$

início
- Inicializar a política com parâmetros θ;
- Inicializar o crítico com parâmetros ϕ;
- **enquanto** *Verdadeiro* **faça**
  - Gerar um episódio $s_0, a_0, r_1, \ldots, s_{T-1}, a_{T_1}, r_T$ de acordo com $\pi(\theta)$;
  - **para** $t = 0, \ldots, T-1$ **faça**
    - $\theta \leftarrow \theta + \alpha \nabla_\theta \log(\pi(a_t|s_t,\theta))(r + \gamma V_\phi(s_{t+1}) - V_\phi(s_t))$;
    - $\phi = \phi - \beta \nabla_\phi (r + \gamma V_\phi(s_{t+1}) - V_\phi(s_t))$;
  - **fim**
- **fim**

fim

Arquitetura

Existem diversos algoritmos que se utilizam da técnica do ator-crítico, entre eles: A2C, A3C, DDPG e SAC [4, 5, 6].

A escolha do tipo de rede ainda depende do problema, podendo ser uma rede densa, convolucional, atenção, entre outras.

Ainda, as redes do crítico e do ator, podem ou não compartilhar camadas.

Limitações dos Métodos Ator-Crítico:

1) Alta Variância: Pode ocorrer devido a recompensas escassas ou ruidosas.

2) Convergência Lenta: Não utiliza um modelo do ambiente, tornando a convergência mais lenta.

3) Erro de Aproximação de Função: Redes neurais aproximam a política e a função de valor, sujeitas a erros.

4) Sensibilidade aos Hiperparâmetros: Escolha dos hiperparâmetros é crucial e difícil.

5) Não Estacionariedade: Mudanças nas probabilidades de transição e recompensas dificultam a aprendizagem.

Métodos ator-crítico na negociação de ações

Métricas de Desempenho:

- Retorno Cumulativo: Indica os retornos totais do investimento.
- Volatilidade Anualizada: Mede a variação nos retornos de um investimento ao longo do tempo, sendo uma métrica de risco.
- Índice de Sharpe: Avalia o desempenho de uma carteira, representando os retornos excedentes por unidade de risco total.
- Máximo Drawdown: Indica a maior perda histórica em um balanço, sendo um indicador de risco.
- Razão Ômega: Métrica de risco-retorno para superar deficiências do Índice de Sharpe.
- Razão Sortino: Calcula o retorno ajustado ao risco, penalizando retornos abaixo de um objetivo específico.

Métodos ator-crítico na negociação de ações

Metric	A2C	DDPG	PPO	TD3	SAC	DJIA
Cumulative return	31.25%	51.97%	58.26%	40.53%	39.42%	22.50%
Annual Volatility	22.86%	20.17%	21.2%	24.57%	19.72%	24.70%
Sharpe ratio	0.53	0.58	0.60	0.54	0.49	0.39
Max drawdown	-28.44%	-33.10%	- 33.22%	- 40.02%	-33.07%	-37.08%
Omega ratio	1.14	1.16	1.16	1.15	1.13	1.08
Sortino ratio	0.75	0.80	0.86	0.79	0.67	0.54

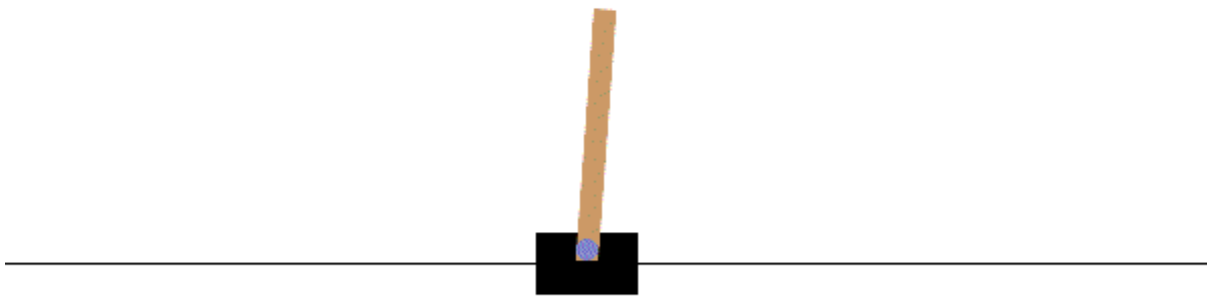
Advantage Actor Critic (A2C), Deep Deterministic Policy Gradient (DDPG), Twin Delayed DDPG (TD3), Proximal Policy Optimization (PPO) e Soft Actor Critic (SAC) [7].
Dow Jones Industrial Average (DJIA) é o Benchmark.

Perguntas?

Exemplo: Cart-Pole

Exemplo TensorFlow

Implementação



Exercícios

[Exercícios Ator-Crítico]

Referências

[1]https://medium.com/@thechrisyoon/deriving-policy-gradients-and-implementing-reinforce-f887949bd63

[2]https://spinningup.openai.com/en/latest/spinningup/rl_intro3.html#id16

[3]https://medium.com/intro-to-artificial-intelligence/the-actor-critic-reinforcement-learning-algorithm-c8095a655c14

[4] https://lilianweng.github.io/posts/2018-04-08-policy-gradient/

[5]https://dilithjay.com/blog/actor-critic-methods

[6]https://medium.com/geekculture/a-deep-dive-into-the-ddpg-algorithm-for-continuous-control-2718222c333e

[7] F. Khemlichi, H. E. Elfilali, H. Chougrad, S. E. Ben Ali and Y. Idrissi Khamlichi, "Actor-Critic Methods in Stock Trading : A Comparative Study," 2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), Tenerife, Canary Islands, Spain, 2023, pp. 1-5, doi: 10.1109/ICECCME57830.2023.10253277.

Obrigado!