Unyleya
EDUCACIONAL

INTRODUÇÃO A APRENDIZAGEM Q

INTRODUÇÃO A APRENDIZAGEM Q

Introdução

Qual a diferença entre um bebê e um adulto?

Por que não nascemos com essas habilidades?

Humanos têm uma longa fase de desenvolvimento antes de atingir a maturidade.

Permite ter a capacidade de nos adaptarmos a uma extensão maior para qualquer ambiente em que nos encontremos: nascemos com alguns instintos específicos e muito espaço para aprendizagem.

Montantes mais perceptíveis de aprendizagem ocorram durante nosso estágio de desenvolvimento, estamos constantemente a afinar nossas habilidades e conhecimentos até morrermos.

INTRODUÇÃO A APRENDIZAGEM Q

Introdução

O meio ambiente, um corpo do próprio organismo, está sempre mudando de forma sutil, um organismo deve ser capaz de reajustar constantemente as suas circunstâncias.

Pense sobre isso também: os organismos vivos são tão complexos que seria impossível para o DNA codificar cada possível sequência de ações ou habilidades que o organismo poderia precisar.

Codifica nossos instintos básicos e, quando aprendemos, ajustamos os instintos em nossos músculos, cérebro etc. de acordo com nossas necessidades específicas.

INTRODUÇÃO A APRENDIZAGEM Q

Introdução

A **aprendizagem por reforço** ajuda a ajustar nossas ações físicas e habilidades motoras.

Ações que um organismo executa resultam em um *feedback*, traduzido em **recompensa** positiva ou negativa por essa ação.

Exemplo: um bebê aprendendo a andar

- Cair e sentir um pouco de dor. *Feedback* **negativo** ajudará o bebê a aprender o que não fazer;
- Capaz de ficar em pé por uma questão de tempo, fazendo algo certo. *Feedback* **positivo**;
- Continua tentando andar, desenvolverá coordenação motora de forma que a **recompensa** será **maximizada**. Dor, fome, sede e prazer são alguns exemplos de reforços naturais.

Ações podem resultar em **recompensas imediatas** ou uma cadeia mais longa de **ações** que levam à **recompensa**.

INTRODUÇÃO A APRENDIZAGEM Q

Q-*Learning*

É um tipo específico de aprendizagem por reforço que atribui **valores a pares de estado-ação**.

O estado do organismo é uma soma de todos os seus dados sensoriais, incluindo sua posição corporal, sua localização no ambiente, a atividade neural em sua cabeça etc.

Em *Q-Learning:* para cada estado há um **número de ações possíveis** que poderiam ser tomadas, cada **ação dentro de cada estado** tem um **valor** de acordo com quantas ou quão poucas recompensas o organismo obterá por completar aquela ação.

Duas formas básicas de aprendizagem Q: **aprendizagem Q ótima** e **aprendizagem Q baseada em política**.

Obrigada!

hulianeufrn@gmail.com

Unyleya
EDUCACIONAL

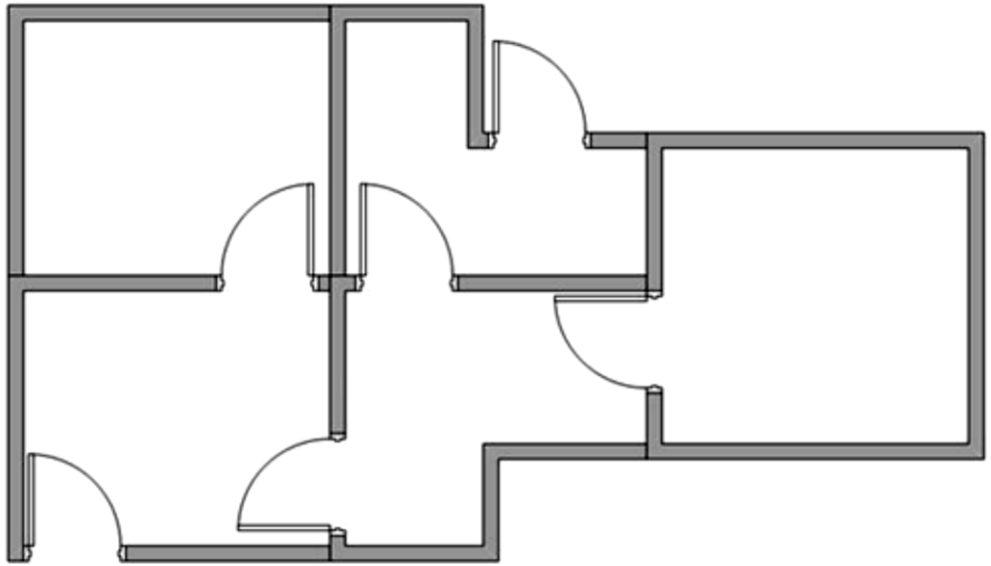
APRENDIZAGEM Q

APRENDIZAGEM Q

Agente usa treinamento não supervisionado para aprender sobre um ambiente desconhecido

Um prédio com 5 salas conectadas por portas. Numerar cada sala de 0 a 4. A parte externa do prédio pode ser considerada uma grande sala. Observe que as portas 1 e 4 levam a parte externa do prédio.

Figura 1: Representação de um prédio e suas portas.



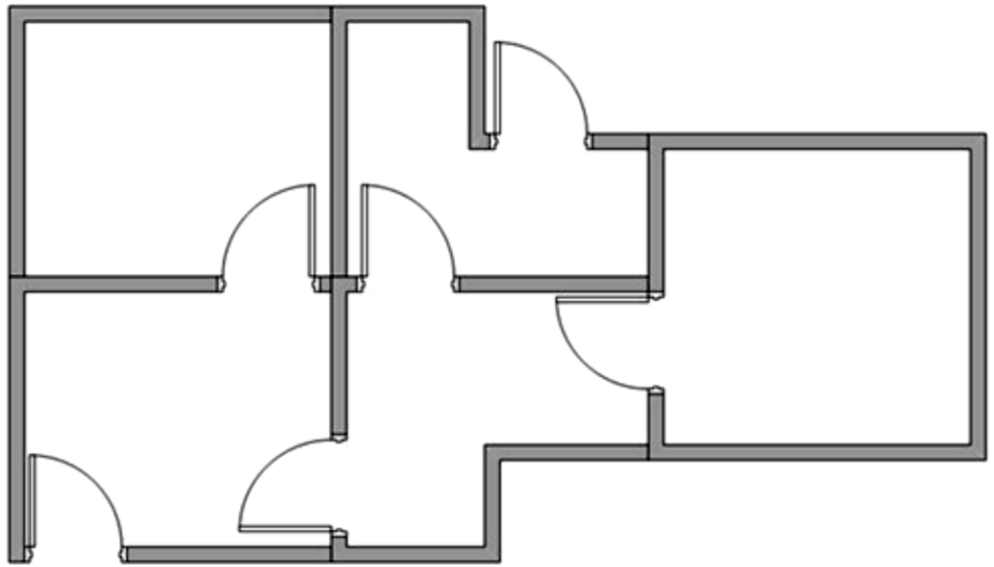
Fonte: McCullock (2012).

APRENDIZAGEM Q

Agente usa treinamento não supervisionado para aprender sobre um ambiente desconhecido

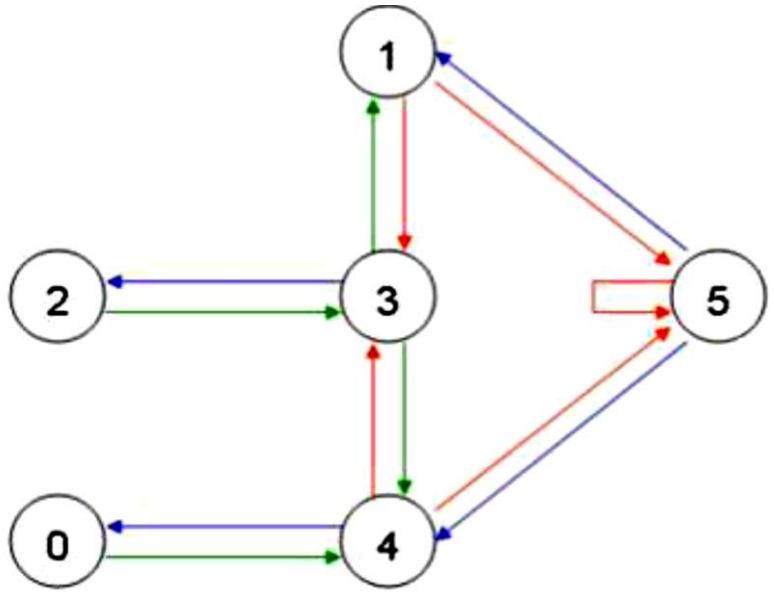
Podemos representar as salas como um grafo. Cada nó representa uma sala e cada porta é representada por um *link*.

Figura 1: Representação de um prédio e suas portas.



Fonte: McCullock (2012).

Figura 2: Nós e portas representados por um link.



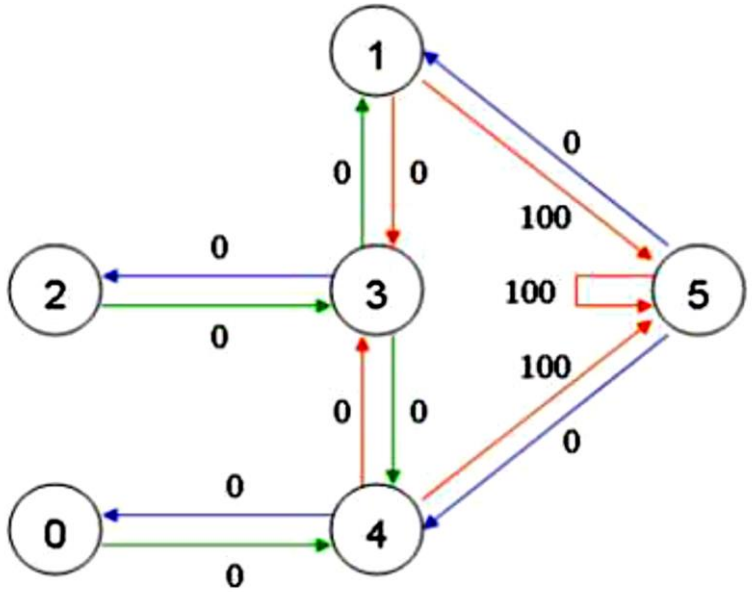
Fonte: McCullock (2012).

APRENDIZAGEM Q

Agente usa treinamento não supervisionado para aprender sobre um ambiente desconhecido

Cada seta contém um valor imediato de recompensa.

Figura 3: Grafo contendo um valor imediato de recompensa.



Fonte: McCullock (2012).

APRENDIZAGEM Q

Agente usa treinamento não supervisionado para aprender sobre um ambiente desconhecido

Na aprendizagem Q (Q-*Learning*), o **objetivo** é **atingir o estado com a maior recompensa**, de modo que se o agente chega ao objetivo, ele permanecerá lá para sempre. Esse tipo de meta é chamado de “absorbing goal” (objetivo absorvente).

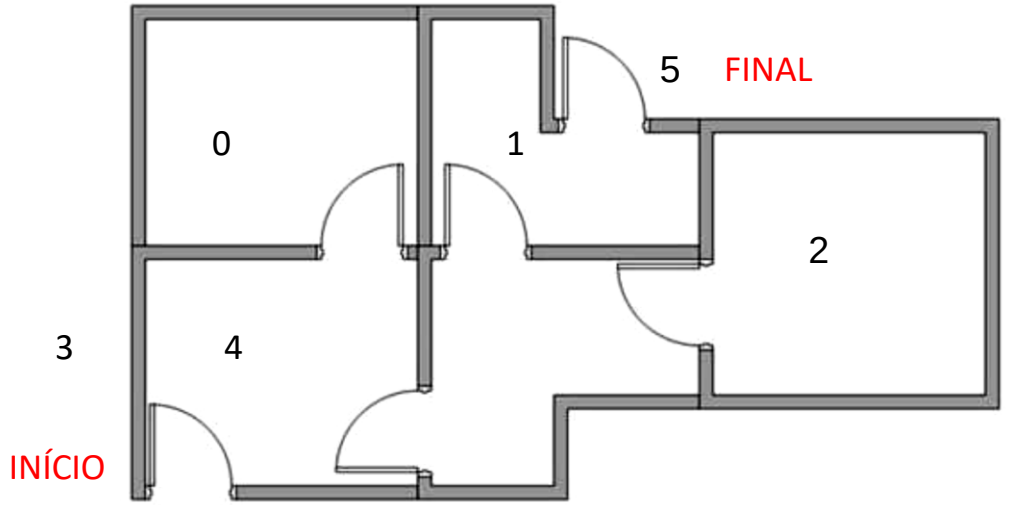
Situação: imagine o nosso agente como um robô virtual “burro” que pode aprender com a experiência. O agente pode passar de uma sala para outra, mas não tem conhecimento do ambiente e não sabe qual sequência de portas leva ao exterior.

APRENDIZAGEM Q

Agente usa treinamento não supervisionado para aprender sobre um ambiente desconhecido

Queremos modelar algum tipo de evacuação simples de um agente, a partir de qualquer sala do prédio. Suponha que temos um agente na sala 2 e queremos que o agente aprenda a chegar até a parte externa do prédio (5).

Figura 4: Agente saindo da ala 2 até a parte externa do prédio (5).



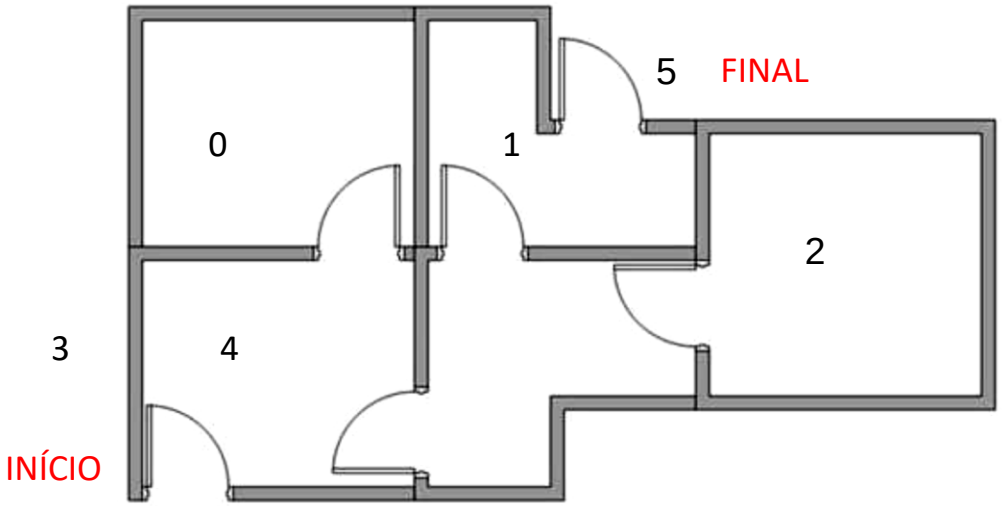
Fonte: McCullock (2012).

APRENDIZAGEM Q

Agente usa treinamento não supervisionado para aprender sobre um ambiente desconhecido

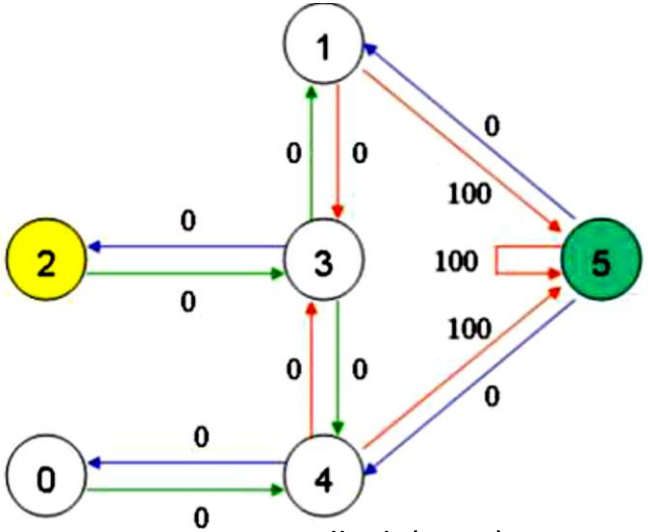
Vamos ligar para cada sala, incluindo a parte externa, um “estado” (representado como um nó), e o movimento do agente de uma sala para outra será uma “ação” (representada pelas setas).

Figura 4: Agente saindo da ala 2 até a parte externa do prédio (5).



Fonte: McCullock (2012).

Figura 5: Ligando cada sala a estados ação.



Fonte: McCullock (2012).

APRENDIZAGEM Q

Agente usa treinamento não supervisionado para aprender sobre um ambiente desconhecido

Suponha que o agente esteja no estado 2.

No estado 2, ele pode ir para o estado 3 porque o estado 2 está conectado ao 3.

No estado 2, no entanto, o agente não pode ir diretamente para o estado 1 porque não há porta direta conectando a sala 1 e 2.

Do estado 3, ele pode ir para o estado 1 ou 4 ou vice-e-versa (observe todas as setas sobre o estado 3).

Se o agente estiver no estado 4, então as três ações possíveis devem ser: ir para o estado 0, 5 ou 3. Se o agente estiver no estado 1, ele pode ir para o estado 5 ou 3. No estado 0, ele só pode ir de volta ao estado 4.

APRENDIZAGEM Q

Agente usa treinamento não supervisionado para aprender sobre um ambiente desconhecido

Podemos colocar o diagrama de estados e os valores imediatos de recompensa numa tabela de recompensas, que chamaremos de “matriz R”. Os “-1s” representam os valores nulos (não há um link entre os nós). Por exemplo, o estado 0 não pode ir para o estado 1.

Figura 6: Matriz “R”.

$$R = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix} \end{array}$$

Fonte: McCullock (2012).

APRENDIZAGEM Q

Matriz similar Q: memória do que o agente aprendeu pela experiência

As linhas representam o estado atual do agente e as colunas representam as possíveis ações que levam ao próximo estado.

O agente não sabendo nada, a matriz Q é inicializada em zero. O número de estados é conhecido (seis). Caso contrário, poderia começar com apenas um elemento. Adiciona mais colunas e linhas na matriz Q se um novo estado for encontrado.

A regra de transição da aprendizagem Q é uma fórmula muito simples:

Q(estado, ação) = R(estado, ação) + Gama * Max [Q(próximo estado, todas as ações)]

Agente robô irá aprender pela experiência. Irá explorar estado por estado até atingir o objetivo.

APRENDIZAGEM Q

Referências

McCULLOCK, J. Q-Learning. 2012. Disponível em: <http://mnemstudio.org/pathfinding-q-learning-tutorial.htm>. Acesso em: 14 set. 2020.

Obrigada!

hulianeufrn@gmail.com

Unyleya
EDUCACIONAL

FUNÇÃO Q

FUNÇÃO Q

A função Q, ou função ação-valor, é definida por $Q^{\pi}: S\ x\ A \rightarrow R$,

$$Q^{\pi}(s_t, a_t) = E_{s_{t+1}} \sim f^{\sim}(s_t, a_t)\ [r(s_t, a_t, s_{t+1}) + \gamma U^{\pi}(s_{t+1})]$$

A partir dessa definição e a da equação de Bellman, observa-se que $Q^{\pi}\ (S_t, \pi\ (S_t,))\ = U_{\pi}(S_t)$, e portanto, a função pode ser expressa na forma de Bellman como:

$$Q^{\pi}(s_t, a_t) = E_{s_{t+1}} \sim f^{\sim}(s_t, a_t)\ [r(s_t, a_t, s_{t+1}) + \gamma Q^{\pi}(s_{t+1}, \pi(s_{t+1}))]$$

FUNÇÃO Q

A função ótima deve satisfazer:

$$Q^*(s_t, a_t) = E_{s_{t+1} \sim f}(s_t, a_t)\,[r(s_t, a_t, s_{t+1}) + \gamma U^*(s_{t+1})]$$
$$= E_{s_{t+1} \sim f}(s_t, a_t)\,[r(s_t, a_t, s_{t+1}) + \gamma Q^*(s_{t+1}, \pi^*(s_{t+1}))]$$

Uma equação da otimalidade de Bellman para funções *Q* é dada por:

$$Q^*(s_t, a_t) = E_{s_{t+1} \sim f}(s_t, a_t)\,[r(s_t, a_t, s_{t+1}) + \gamma \max_{a \in U} Q^*(s_{t+1}, a)]$$

FUNÇÃO Q

Essa equação caracteriza Q, e declara que o valor ótimo de uma ação a_t aplicada no estado s_t é o valor esperado da soma da recompensa imediata com o valor ótimo descontado obtido pela melhor ação no estado seguinte.

Uma política ótima π pode ser determinada a partir de Q por:

$$\pi^*(s_t) = argmax_a Q^*(s_t, a)$$

FUNÇÃO Q

A função Q depende também da ação, ela já inclui informação sobre a qualidade das transições.

Por outro lado, a função valor de estado U somente descreve a qualidade dos estados.

Para inferir a qualidade das transições, essas devem ser explicitamente levadas em consideração.

Um modelo do MDP é necessário na forma da dinâmica e $f \sim$ da função de utilidade r, enquanto que na formulação usando função Q, o problema de decisão markoviano é tratado sem referência a tais modelos.

Obrigada!

hulianeufrn@gmail.com

Unyleya
EDUCACIONAL

APRENDIZAGEM Q ÓTIMA

APRENDIZAGEM Q ÓTIMA

A **aprendizagem Q** é um método de política-*off*. A função valor ótima é estimada, independentemente da política atual (exploração) que está sendo utilizada para gerar as trajetórias amostradas.

A regra de aprendizagem Q ótima pode ser expressa como:

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha_t[r(s_t, a_t, s_{t+1}) + \gamma max_{a \epsilon U} Q_t(s_{t+1}, a) - Q_t(s_t, a_t)]$$

sendo Q_{t+1} *a t-ésima* estimativa para Q^* e a_t os tamanhos de passos (*stepsizes*).

APRENDIZAGEM Q ÓTIMA

Aprendizagem Q converge com probabilidade 1 para o valor ótimo Q^* quando $t \rightarrow \infty$, , sob as seguintes suposições, (WATKINS e DAYAN, 1992):

1. uma sequência decrescente apropriada de tamanhos dos passos que satisfaz as condições $\Sigma_{t=0}^{\infty} \propto_t = \infty$ e $\Sigma_{t=0}^{\infty} \propto_t^2 < \infty$ e
2. todos os pares estado-ação são visitados, assintoticamente, infinitas vezes.

Por exemplo, uma sequência padrão de tamanhos de passos que garante a convergência da aprendizagem Q é dada por:

$$\alpha_t = \frac{\tau}{\eta + t}, \; t = 1, 2, \ldots,$$

sendo τ e η números positivos.

APRENDIZAGEM Q ÓTIMA

A condição 2, pode ser satisfeita se o agente de aprendizagem selecionar com probabilidade não nula uma ação aleatória em cada estado visitado (exploração).

Além disso, exploitar seu conhecimento atual para obter um bom desempenho, por exemplo, selecionado ações gulosas com respeito à sua função Q atual (exploitação).

Uma abordagem característica que relaciona-se com exploração-exploitação em algoritmos de aprendizagem por reforço é chamada exploração gulosa (SUTTON e BARTO, 1998).

Seleciona ações de acordo com a seguinte regra: $a_{t} = a \in argmax_{a}\, Q_t(S_t, a)$, com probabilidade $1 - \varepsilon_t$ uma ação uniformemente aleatória em U (), com probabilidade sendo ε_t $(0, 1)$ a probabilidade de exploração no passo de tempo t.

Obrigada!

hulianeufrn@gmail.com

Unyleya
EDUCACIONAL

APRENDIZAGEM BASEADA EM POLÍTICA

APRENDIZAGEM BASEADA EM POLÍTICA

A **aprendizagem Q baseada em política** tenta aprender, a função Q para alguma política projetada.

A política pode ser ou não a política que de fato está sendo seguida durante o treinamento. A regra de aprendizagem baseada em política é dada por:

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha_t[r(s_t, a_t, s_{t+1}) + \gamma Q_t(s_{t+1}, \pi(s_{t+1})) - Q_t(s_t, a_t)]$$

Sendo Q_t a *t-ésima* estimativa para Q_π.

APRENDIZAGEM BASEADA EM POLÍTICA

Trabalhos na área

Pouco depois dos trabalhos de Watkins e Dayan (1992) e Watkins (1989), Peng e Williams (1996) apresentaram um método de aprendizagem Q multi-passos incremental $Q(\lambda)$, uma combinação de aprendizagem Q a um passo e $TD(\lambda)$.

Sutton e Barto (2018) fizeram uma comparação entre aprendizagem Q multi-passos, $Q(\lambda)$ e o método sugerido por Watkins (1989).

Similarmente, Kuzmin (2002) sugeriu uma modificação à aprendizagem Q tabular tradicional. Propõe um *perceptron* multicamadas para atuar como um aproximador de avaliação de aprendizagem Q. Denominou essa combinação de “conexionismo”.

APRENDIZAGEM BASEADA EM POLÍTICA

Trabalhos na área

Um esquema similar para aproximação de função valor de estado foi proposto por Pipe (1998), onde o método tabular tradicional foi substituído por uma aproximação de função via uma função de base radical. Denominou a função valor de estado como um campo potencial.

Deng e Er (2004) propuseram a aprendizagem Q *Fuzzy*. A aprendizagem Q é usada para gerar e ajustar automaticamente regras *fuzzy* para uma tarefa de navegação de robô móvel.

Bertsekas e Yu (2010) propuseram aprendizagem Q e melhoraram a iteração de política em programação dinâmica com desconto.

APRENDIZAGEM BASEADA EM POLÍTICA

Referências

WATKINS, C. J. C. H. Learning from delayed rewards. PhD thesis. University of Cambridge. Cambridge, England, 1989.

WATKINS, C. J. C. H,; DAYAN, P. Q-learning. In: Machine Learning. Vol. 8, pp. 279- 292, 1992.

PENG, J.; WILLIAMS, R. J. Incremental multi-step Q-learning. In: Machine Learning. Morgan Kaufmann, pp. 226-232, 1996.

SUTTON, R. S.; BARTO, A. G. Reinforcement learning: an introduction, 2ªEd. MIT Press, Cambridge, Massachusetts, EUA, 2018.

KUZMIN, V. Connectionist Q-learning in robot control task. 2002.

PIPE, A. G. An architecture for building “potential field” cognitive maps in mobile robot navigation. In: Systems, Man and Cybernetics, IEEE International Conference on. Vol. 3. pp. 2413-2417. 1998.

DENG, C.; ER, M. J. Real-time dynamic fuzzy Q-learning and control of mobile robots. In: Control Conference, 5th Asian. Vol. 3. pp. 1568-1576. 2004.

BERSEKAS, D. P.; YU. H. Q-learning and enhanced policy iteration in discounted dynamic programming. In: Decision and Control (CDC), 2010, 49th, IEEE Conference on. pp. 1409-1416.

Obrigada!

hulianeufrn@gmail.com

Unyleya
EDUCACIONAL

ALGORITMO Q-*LEARNING*

ALGORITMO Q-*LEARNING*

Passos do algoritmo

1. defina o parâmetro gama e as recompensas do ambiente na matriz R;
2. inicialize a matriz Q com zero;
3. para cada episódio:
    a. selecione um estado inicial aleatório;
    b. enquanto o estado objetivo não for alcançado:
        I. selecione uma entre todas as ações possíveis para o estado atual;
        II. usando essa ação possível, considere ir para o próximo estado;
        III. obtenha o valor Q máximo para este próximo estado com base em todas as ações possíveis;
        IV. calcule: Q(estado, ação) = R(estado, ação) + Gama * Máx [Q(próximo estado, todas as ações)];
        V. defina o próximo estado com o estado atual.

ALGORITMO Q-*LEARNING*

O algoritmo é empregado pelo agente para aprender com a experiência

Cada episódio é equivalente a uma sessão de treinamento.

Cada sessão de treinamento, o agente explora o ambiente recebe a recompensa até atingir o estado objetivo.

O objetivo do treinamento é melhorar o “cérebro” do nosso agente, representado pela matriz Q.

Mais treinamento resulta em uma matriz mais otimizada Q. Matriz Q aprimorada, o agente encontrará a rota mais rápida para o estado objetivo.

O gama ([0,1]) mais próximo de zero, o agente tenderá a considerar recompensas imediatas. Gama mais próximo de um, o agente considerará recompensas futuras com maior peso, disposto a atrasar a recompensa.

ALGORITMO Q-*LEARNING*

Passos do algoritmo para utilizar a matriz Q

O agente rastreia a sequência de estados, do estado inicial ao estado objetivo. O algoritmo localiza as ações com os maiores valores de recompensa registrados na matriz Q para o estado atual.

1. defina o estado atual = estado inicial;
2. a partir do estado atual, localize a ação com maior valor Q;
3. defina o estado atual = próximo estado;
4. repita as etapas 2 e 3 até o próximo estado atual = estado objetivo.
5. O algoritmo acima retornará a sequência de estados do estado inicial para o estado
6. objetivo.

O algoritmo acima retornará a sequência de estados do estado inicial para o estado objetivo.

ALGORITMO Q-*LEARNING*

Exemplo algoritmo Q-*Learning*

Começaremos definindo o valor do parâmetro de aprendizado gama = 0.8 e o estado inicial como Sala 1.

Figura 1: Inicialização da Matriz Q.

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Fonte: McCullock (2012).

Figura 2: Matriz R.

$$R = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix} \end{array}$$

Fonte: McCullock (2012).

Segunda linha (estado 1) da matriz R existem duas ações possíveis para o estado atual 1: vá para o estado 3 ou vá para o estado 5. Seleção aleatória, iremos para o estado 5 como nossa ação.

ALGORITMO Q-*LEARNING*

O que aconteceria se nosso agente estivesse no estado 5?

Na sexta linha da matriz de recompensa R (estado 5), temos três ações possíveis: estado 1, 4 ou 5.

Q(estado, ação) = R(estado, ação) + Gama * Máx [Q(próximo estado, todas as ações)]

Q(1, 5) = R(1, 5) + 0.8 * Máx [Q(5, 1), Q(5, 4), Q(5, 5)] = 100 + 0.8 * 0 = 100

O resultado de Q(1, 5) é 100. O próximo estado, 5, agora se torna o estado atual (estado objetivo).

Figura 3: Matriz atualizada Q do agente.

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Fonte: McCullock (2012).

ALGORITMO Q-*LEARNING*

Para o próximo episódio, começamos com um estado inicial escolhido aleatoriamente

Estado = 3 como nosso estado inicial. Na quarta linha da matriz R, o agente tem 3 ações possíveis: 1, 2 ou 4. Aleatoriamente, iremos para o estado 1 como nossa ação. No estado 1 da matriz de recompensa R, tem duas ações possíveis: 3, 5.

Q(estado, ação) = R(estado, ação) + Gama * Máx [Q(próximo estado, todas as ações)]

Q(1, 5) = R(1, 5) + 0.8 * Máx [Q(1, 2), Q(1, 5)] = 0 + 0.8 * Máx(0, 100) = 80

Figura 4: Matriz Q atualizada.

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 80 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

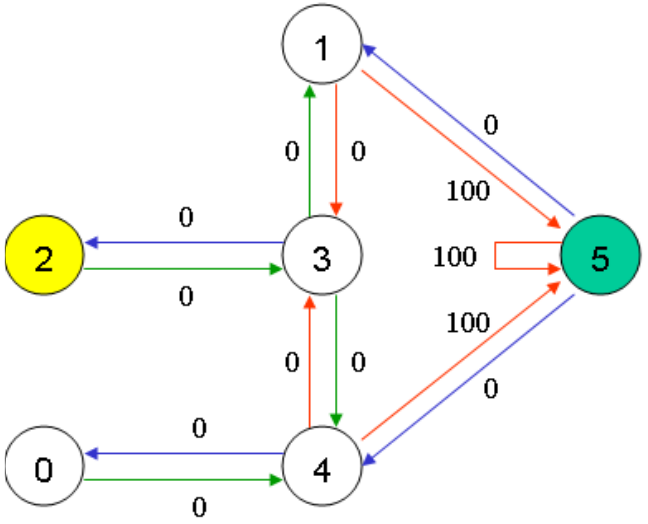
Fonte: McCullock (2012).

ALGORITMO Q-*LEARNING*

O próximo estado, 1, agora se torna o estado atual

Repetimos o *loop* interno do algoritmo de aprendizado Q porque o estado 1 não é o estado do objetivo. Iniciando o novo *loop* com o estado atual 1, há duas ações possíveis: 3, 5. Aleatoriamente, ação 5.

Figura 5: Novo *loop* com estado atual 1.



Fonte: McCullock (2012).

ALGORITMO Q-*LEARNING*

Agora que estamos no estado 5

Há três ações possíveis: ir para o estado 1, 4 ou 5.

Q(estado, ação) = R(estado, ação) + Gama * Máx [Q(próximo estado, todas as ações)]
Q(1, 5) = R(1, 5) + 0.8 * Máx [Q(5, 1), Q(5, 4), Q(5, 5)] = 100 + 0.8 * 0 = 100

O resultado desse cálculo para Q (1, 5) é 100 por causa da recompensa instantânea de R (5, 1). Como 5é o estado objetivo, terminamos este episódio.

Figura 6: Matriz Q atualizada

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 400 & 0 \\ 0 & 0 & 0 & 320 & 0 & 500 \\ 0 & 0 & 0 & 320 & 0 & 0 \\ 0 & 400 & 256 & 0 & 400 & 0 \\ 320 & 0 & 0 & 320 & 0 & 500 \\ 0 & 400 & 0 & 0 & 400 & 500 \end{bmatrix} \end{array}$$

Fonte: McCullock (2012).

ALGORITMO Q-*LEARNING*

Se o nosso agente aprender mais com mais episódios

Ele finalmente alcançará os valores de convergência na matriz Q. Esta matriz Q pode então ser normalizada (isto é, convertida em porcentagem) dividindo todas as entradas diferentes de zero pelo número mais alto (500 neste caso)

Figura 7: Convergência na matriz Q.

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 80 & 0 \\ 0 & 0 & 0 & 64 & 0 & 100 \\ 0 & 0 & 0 & 64 & 0 & 0 \\ 0 & 80 & 51 & 0 & 80 & 0 \\ 64 & 0 & 0 & 64 & 0 & 100 \\ 0 & 80 & 0 & 0 & 80 & 100 \end{bmatrix} \end{array}$$

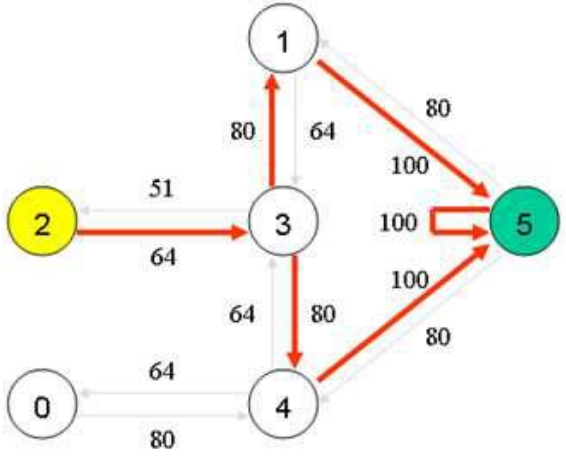
Fonte: McCullock (2012).

ALGORITMO Q-*LEARNING*

Matriz Q se aproximar o suficiente de um estado de convergência

O agente aprendeu os melhores caminhos para o estado objetivo. Rastrear as melhores sequências de estados é tão simples quanto seguir os *links* com os valores mais altos em cada estado.

Figura 8: Rastreando as melhores sequências de estados.



Estado “objetivo”

Fonte: McCullock (2012).

Assim, a sequência é 2 - 3 - 1 - 5.

ALGORITMO Q-*LEARNING*

Referência

McCULLOCK, J. **Q-Learning.** 2012. Disponível em: <http://mnemstudio.org/pathfinding-q-learning-tutorial.htm>. Acesso em: 15 set. 2020.

Obrigada!

hulianeufrn@gmail.com

Unyleya
EDUCACIONAL

DETALHES DO ALGORITMO Q-*LEARNING*

DETALHES DO ALGORITMO Q-LEARNING

Algoritmo Q-*Learning* definido por Watkins (1989)

Figura 1: Inicialização da Matriz Q.

Algoritmo Q-Learning

procedure QLearning(r, α_q, ε, γ)
Inicialize $Q(s, a)$
    repeat
        Inicialize s
        repeat
            Selecione α de acordo com a política ε-gulosa
            Observe os valores de r e s'
            $Q(s, a) \leftarrow (1 - \alpha_q)\ Q(s, a) + \alpha_q(r + \gamma max_{a \in A}(Q(s', a)))$
            $s \leftarrow s'$
        until Encontrar um estado final
    until Atingir $N \in p$ episódios
    return $Q(s, a)$ *{Matriz dos Q-valores}*
end procedure

Fonte: adaptado de Watkins (1989)

DETALHES DO ALGORITMO Q-LEARNING

Políticas de seleção de ações para o algoritmo *Q-Learning*

Uma política de seleção de ações tem como objetivo estabelecer o comportamento do agente aprendiz para que ele alterne adequadamente entre o uso do conhecimento já adquirido e a aquisição de novo conhecimento.

Otimizar o processo de exploração/exploitação do espaço de busca.

A ideia de experimentar mais de uma política de seleção de ações para o Q-*Learning,* tem como meta verificar qual dessas políticas é mais adequada para ser utilizada na implementação dos métodos híbridos propostos.

DETALHES DO ALGORITMO Q-LEARNING

Política ε – gulosa

Escolhe a ação que tem o maior valor esperado, com probabilidade definida por $(1 - \varepsilon)$, e de ação aleatória, com probabilidade ε. Dada a matriz Q-valores Q obtém-se a ação gulosa para um estado s fazendo:

$$a^* = max_{a \in A(s)} Q(s, a)$$

$$\pi(s, a^*) = 1 - \varepsilon + \frac{\varepsilon}{|A(s)|}$$

$$\pi(s, a) = \frac{\varepsilon}{|A(s)|} \forall_{a \in A(s)} - \{a^*\}$$

Onde $|A(s)|$ corresponde ao número de ações possíveis de serem executadas a partir de s, e ε é o parâmetro de controle entre gula e aleatoriedade. A restrição presente permite que o Q-*Learning* explore o espaço de estados do problema, e é uma das condições necessárias encontrar uma política de controle ótima.

DETALHES DO ALGORITMO Q-LEARNING

Política ε - gulosa adaptativa

Semelhante à política gulosa descrita anteriormente. Permite escolher a ação que tem o maior valor esperado, com probabilidade definida por (1 - ε), e ação aleatória, com probabilidade .

Diferença e justifica o termo “adaptativa” é que o valor sofre um decaimento exponencial calculado por:

$$\varepsilon = max\{v_i,\ v_f.b^k\}$$

Onde k é o contador de episódios do Q-*Learning,* b é um valor próximo de 1 e $v_i < v_f \in [0,1]$.

O algoritmo utilizará valores grandes ε, e à medida que o valor de k cresce a escolha de b é direcionada para valores menores. A ideia é permitir que sejam feitas escolhas mais aleatórias e, à medida que o número de episódios aumente, o especto guloso seja mais explorado.

DETALHES DO ALGORITMO Q-LEARNING

Política baseada na contagem de visitas

A escolha de ações é feita baseada em uma técnica denominada Comparação de Reforço (*Reinforcement Comparison*) (SUTTON e BARTO, 2018).

Princípio de que ações seguidas de grandes recompensas devem ser preferidas em detrimento de ações seguidas de pequenas recompensas.

“Grande recompensa” compara com um nível de recompensa padrão denominado recompensa referencial.

A ideia semelhante à técnica de comparação de reforço. A escolha das ações preferidas é feita com base na contagem de visitas aos estados atingidos por tais ações.

DETALHES DO ALGORITMO Q-LEARNING

Política baseada na contagem de visitas

Medida de preferência de ações, determina a probabilidade de seleção das ações de acordo com a seguinte relação:

$$\pi_t(a) = P_r\{a_t = a\} = \frac{e^{p_{t-1}(a)}}{\sum_b e^{p_{t-1}(b)}}$$

Onde $\pi_t(a)$ denota a probabilidade de se escolher a ação a no passo t e $p_t(a)$ denota a preferência da ação a no tempo t, que é calculada por:

$$p_{t+1}(a_t) = p_t(a_t) + \beta(N_v(s, a_t) / NE_p)$$

Onde s é o estado atingido em consequência da escolha da ação a no passo $t, N_v(s, \alpha t)$ é o número de visitas ao estado s, N_{vp} é o número total de episódios e $\beta \in [0, 1]$ é um parâmetro de controle que pondera o nível de influência das ações preferenciais.

DETALHES DO ALGORITMO Q-LEARNING

Determinação da função de recompensa

Um agente aprendiz tem como objetivo maximizar o total de recompensa recebida ao longo do processo. O agente necessita mensurar quão bom é escolher uma determinada ação a partir do estado corrente por meio de um valor numérico, isso é feito pela função de recompensa.

Para cada problema específico se faz necessário definir a função de recompensa.

A função de recompensa utilizada no algoritmo *Q-Learning*:

$$P_{ss'}^{a} = P_r\{s_{t+1} = s', r_{t+1} = r \mid s_t, a_t\}$$

estabelece de forma trivial a recompensa imediata como sendo inversamente proporcional à distância de um estado de partida à cidade de destino.

DETALHES DO ALGORITMO Q-LEARNING

Determinação da função de recompensa

Esta forma de estabelecer a recompensa imediata apresenta a inconveniência de, em casos e valores de distâncias muito próximos, ocorrer casos de empate na escolha dos Q-valores $Q(s, a)$.

Ponderar o inverso da distância por algum valor que permita distinguir os casos de empate. Utilizando a contagem de visitas aos estados do ambiente.

As ações que levam a estados mais frequentemente visitados serão premiadas com recompensa imediata maior. Isso pode ser feito utilizando a seguinte equação:

$$R(s, a) = \frac{1}{d_{ij}} * N_v(s, a)$$

Onde $\frac{1}{d_{ij}} 1$ é o inverso da distância entre os estados c_i e c_j (sendo o estado representado pelo estado s, e o estado representado pelo estado a ser atingido em consequência da escolha da ação a e $N_v\ (s, a)$ o número de visitas ao estado corrente.

DETALHES DO ALGORITMO Q-LEARNING

Referência

WATKINS, C. J. C. H. Learning from delayed rewards. PhD thesis. University of Cambridge. Cambridge, England, 1989.

SUTTON, R. S.; BARTO, A. G. Reinforcement learning: an introduction, 2ªEd. MIT Press, Cambridge, Massachusetts, EUA, 2018.

Obrigada!

hulianeufrn@gmail.com

Unyleya
EDUCACIONAL

Q-*LEARNING* λ

Q-*LEARNING* λ

Q-*LEARNING* λ (ELIGIBILITY TRACES)

Eligibility traces (ou traços de elegibilidade) apresentado por Klopf (1972) no âmbito de sistemas adaptativos para a USAF (*United States Air Force*).

Em aprendizagem por reforço somente foi realizada em 1989 por Watkins (1989) ao incorporar este no mecanismo de aprendizagem *Q-Learning.*

Elegibility traces consiste no registo temporário da ocorrência de um determinado evento, quer seja a passagem por um estado ou a seleção de uma determinada ação. Registro permite a sinalização de quaisquer parâmetros associados ao evento como elegíveis para posteriormente poderem sofrer alterações na sua definição. Na ocorrência de um erro, este poderá ter a sua origem relacionada com os estados e/ou ações sinalizadas.

Q-*LEARNING* λ

Mecanismo de aprendizagem *Q-Learning* (*elegibility traces*) definido por Watkins (1989)

Figura 1: Algoritmo *Q-Learning*.

1.Inicialize ***Q(s, a)*** arbitrariamente e ***e(s, a)* = 0**; para todo ***s, a***
2. Repita (para cada episódio)
   a) Inicialize ***s***
   b) Escolha ***a*** a partir de ***s*** usando a política derivada de Q
   c) Repita (para cada episódio)
   d)
      i. Tome uma ação ***a***, e observe ***r***, ***s'***
      ii. Escolha ***a'*** a partir de ***s'*** usando a política derivada de Q
      iii. $\delta \leftarrow r + \gamma Q(s', a^*) - Q(s, a)$
      iv. $e(s, a) \leftarrow e(s, a) + 1$
      v. Para todo ***s***, ***a:***
         1. $Q(s, a) \leftarrow Q(s, a) + \alpha\delta e(s, a)$
         2. Se ***a'*** = a^* então
            a. $e(s, a) \leftarrow \gamma\lambda e(s, a)$
         3. else
            a. $e(s, a) \leftarrow 0$
      vi. $s \leftarrow s'$; $a \leftarrow a'$
   e) até o terminal ***s***

Fonte: Watkins (1989).

Q-*LEARNING* λ

Mecanismo de aprendizagem *Q-Learning* (*elegibility traces*) definido por Watkins (1989)

Parâmetro de *fator de rastreamento* [0, 1] que tem uma relação inversa com a *taxa de decaimento de sinalização do evento*.

Os eventos sejam sinalizados por uma função de ação-valor auxiliar *e(s, a)*, essa sinalização é automaticamente incorporada na estrutura única de definição do ambiente.

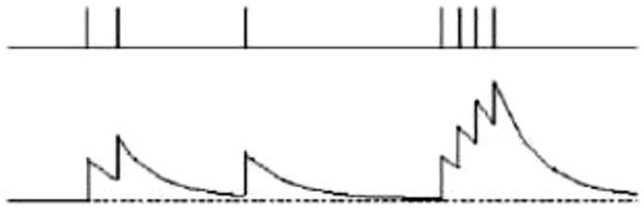
Em cada estado *s* existente, somente são sinalizados os eventos referentes à ação *a* que nesse estado se encontre mais valorizada, sendo as possíveis ações restantes mantidas sem qualquer sinalização.

Q-*LEARNING* λ

Mecanismo de aprendizagem *Q-Learning* (*elegibility traces*) definido por Watkins (1989)

A sinalização de um evento é realizada de forma incremental e unitária, o que significa que eventos da mesma natureza que ocorram próximos no tempo irão incrementar cumulativamente a sua sinalização.

Figura 2: Sinalização de eventos de forma acumulada.



Fonte: Watkins (1989).

Q-*LEARNING λ*

Q-*Learning λ* (*replacing*), apresentado por Sutton e Satinder (1996)

A sinalização de um evento no conceito original ser realizada de uma forma incremental. Eventos da mesma natureza que ocorressem próximos no tempo, seria possível que estes ficassem sinalizados de forma excessiva, e por consequência avaliados pelo processo de aprendizagem.

Alternativa a substituição do processo de sinalização de eventos: atribuição de um valor unitário único, não acumulável, demonstrando ainda que essa alteração se manifesta mais rápida e origina informação de maior qualidade durante o processo de aprendizagem.

Mecanismo de aprendizagem é idêntico à do mecanismo *eligibility traces*, apresentando uma sutil e relevante diferença ao afetar com um valor unitário não acumulável a sinalização de um evento.

Q-*LEARNING* λ

Mecanismo de aprendizagem *Q-Learning* (*replacing*) definida por Sutton e Satinder (1996)

Figura 3: Algoritmo Q-*Learning* λ *(replacing*).

1. Inicialize ***Q(s, a)*** arbitrariamente e ***e(s, a)* = 0**; para todo ***s, a***
2. Repita (para cada episódio)

a) Inicialize ***s***

b) Escolha ***a*** a partir de ***s*** usando a política derivada de Q

c) Repita (para cada episódio)

I. Tome uma ação ***a***, e observe ***r***, ***s'***

II. Escolha ***a'*** a partir de ***s'*** usando a política derivada de Q

III. $\delta \leftarrow r + \gamma Q(s', a^*) - Q(s, a)$

IV. $e(s, a) \leftarrow 1$

V. Para todo ***s, a:***

(1) $Q(s, a) \leftarrow Q(s, a) + \alpha\delta e(s, a)$

(2) Se ***a'*** = a^* então

(a) $e(s, a) \leftarrow \gamma\lambda e(s, a)$

(3) else

(a) $e(s, a) \leftarrow 0$

VI) $s \leftarrow s'$; $a \leftarrow a'$

d) até o terminal ***s***

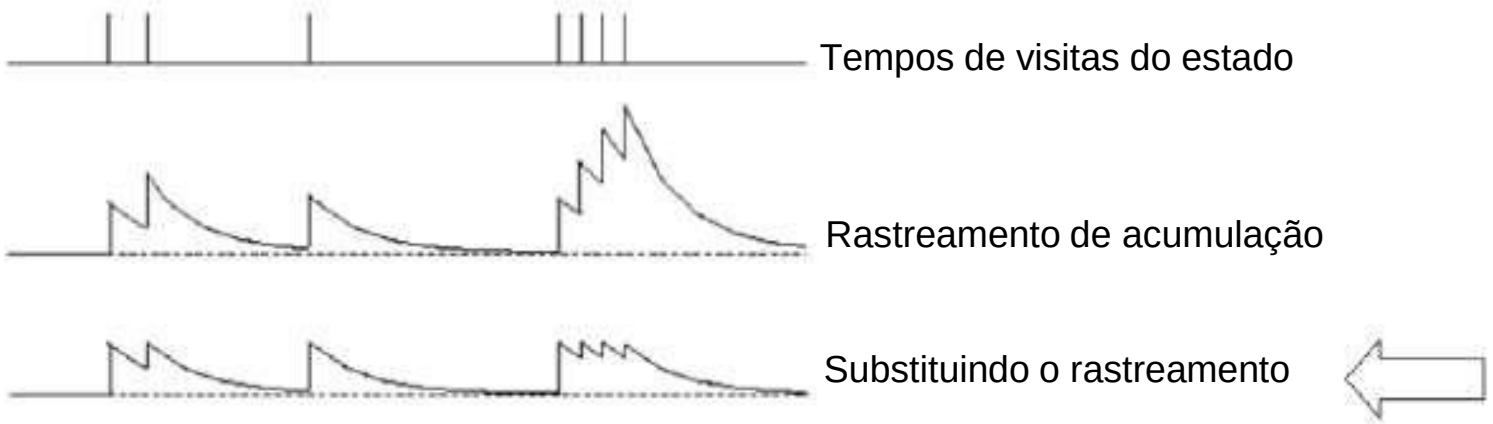
Fonte: Sutton e Satinder 1996).

Q-*LEARNING* λ

Mecanismo de aprendizagem *Q-Learning* (*replacing*) definida por Sutton e Satinder (1996)

A sinalização de um evento realizada de forma unitária e não incremental, eventos da mesma natureza que ocorram próximos no tempo serão sempre sinalizados com um valor máximo e não acumulativo com sinalizações anteriores.

Figura 4 Sinalização de eventos de forma unitária.



Fonte: Sutton e Satinder (1996).

Q-LEARNING λ

Referência

KLOPF, A. H. Brain function and adaptive systems. A heterostatic theory. Bedford, Massachusetts: Air Force Cambridge Research Laboratories, 1972.

WATKINS, C. J. C. H. Learning from delayed rewards. PhD thesis. University of Cambridge. Cambridge, England, 1989.

SUTTON, R. S.; SATINDER, S. P. Reinforcement Learning with Replacing Elegibility Traces. - Cambridge : Dept. of Computer Science, MIT, 1996.

Obrigada!

hulianeufrn@gmail.com

Unyleya
EDUCACIONAL

APLICAÇÕES Q-*LEARNING*

APLICAÇÕES Q-*LEARNING*

Robô humanoide

Os motores do humanoide podem assumir três velocidades: 0, 30 e 50 unidades de potência. São acionados de forma intercalada, por 1 segundo. As ações possíveis são:

- não alterar a potência do motor;
- diminuir ou aumentar a potência.

Quando o robô humanoide identifica um obstáculo, simulando uma presa, o robô começa a andar. O crítico avalia a ação em função deslocamento do robô em relação ao alvo por um período de tempo.

- se o robô se desloca até um limite x_1, a ação é punida;
- se o deslocamento é de x_1 até x_2, a ação é punida com menor severidade;
- finalmente, se o robô se desloca a uma distância maior que x_2, a ação é recompensada.

APLICAÇÕES Q-*LEARNING*

Robô humanoide

A tabela *Q-Learning* tem dimensão N_e x N_a, sendo N_e o número de estados e N_a o número de ações.

Na descrição do exemplo foi dado que há dois motores assumindo três estados e cada motor pode executar três ações. Assim, = 3^2 = 9 e = 3^2 = 9.

O $Q(s, a)$ é ótimo nas velocidades $(50, 50)$, pois são máximas e não desestabilizam o humanoide e ação (0, 0), pois não alteram a velocidade máxima.

APLICAÇÕES Q-*LEARNING*

Tabela 1 – Tabela *Q-Learning* antes e depois do treinamento

Matriz antes de aprender								
0,95	0,01	0,48	0,36	0,49	0,79	0,4	0,18	0,79
0,07	0,71	0,11	0,56	0,7	0,66	0,64	0,88	0,17
0,62	0,37	0,88	0,69	0,95	0,03	0,08	0,14	0,45
0,35	0,14	0,39	0,89	0,75	0,57	0,44	0,78	0,4
0,79	0,12	0,92	0,91	0,43	0,44	0,14	0,01	0,02
0,25	0,91	0,75	0,39	0,08	0,01	0,85	0,65	0,69
0,3	0,34	0,73	0,77	0,61	0,98	0,44	0,46	0,62
0,19	0,02	0,28	0,3	0,9	0,63	0,65	0,17	1
0,17	0,59	0,03	0,01	0,84	0,23	0,75	0,16	0,21
Matriz depois de aprender								
0,48	-0,7	0,42	-0,65	-0,57	0,023	-0,183	0,03	0,32
-0,01	0,46	-0,92	-0,66	0,37	0,6	0,55	0,07	-0,26
-0,69	0,3	0,83	0,22	0,42	-0,21	0,07	-0,2	-0,47
-0,67	-0,535	-0,48	0,37	0,01	0,14	-0,03	0,56	0,19
0,03	-0,78	0,52	0,8	-0,88	-0,78	-0,59	-0,14	0,1
-0,528	0,3	0,11	-0,88	-0,53	-0,03	0,52	0,07	0,23
-0,84	0,13	0,67	0,76	0,523	0,07	-0,52	-0,69	0,012
0,1	-0,33	-0,304	0,28	0,56	0,26	0,64	-0,34	0,1
-0,36	0,21	-0,13	-0,2	8,424	-0,46	0,24	-0,391	-0,368

Fonte: Silva e Gouvêa, (2015).

APLICAÇÕES Q-*LEARNING*

Um Conto de Natal

Jonas é motorista de taxi e gosta de trabalhar na madrugada. Estuda sobre inteligência Artificial e assuntos relacionados. Seu objetivo é conseguir um emprego nessa área.

Interesse por Q-*Learning*, um aprendizado por reforço, e escreveu alguns códigos no tempo livre sobre como ensinar um robô virtual a encontrar a saída de um labirinto com armadilhas.

Enquanto estudava e esperava passageiros, avista um homem era barrigudo, com uma grande barba branca, vestia roupas vermelhas e carregava um embrulho em suas mãos – era o Papai Noel – que adentrou em seu carro e gritou: Preciso de sua ajuda jovem, estamos quase sem tempo. Me leve a este local o mais rápido possível, VAMOS!

Ficou claro que deveria chegar ao destino antes que o sol nascesse.

APLICAÇÕES Q-*LEARNING*

Um Conto de Natal

O problema é que ele conhecia vários caminhos possíveis até o destino final e sua pouca experiência não o permitia saber qual o mais rápido. **Como, então, ele iria descobrir qual o menor caminho até lá?**

Rapidamente pegou seu *notebook* no banco de trás do carro, abriu e começou a digitar algumas coisas. Alguns segundos foram suficientes para que Jonas rodasse um de seus códigos e, com um largo sorriso no rosto, arrancasse com seu carro.– Não se preocupe Papai Noel, não deixarei que isso aconteça.

APLICAÇÕES Q-*LEARNING*

Jonas estabeleceu uma espécie de modelo para a cidade indicando o destino.

Cada número representa um dos lugares em que está o taxi (células). A movimentação permitida é apenas para cima, baixo, esquerda ou direita. As células com um “T” são lugares em que há ruas em obras, já o “X” marca o destino, onde o presente final será entregue.

Figura 1 – Representação de uma cidade

1	5	9	13 T
2	6	10	14
3	7	11	15
4	8 T	12	16 X

Fonte: Cabral (2018).

APLICAÇÕES Q-*LEARNING*

Referência

SILVA, T. L,; GOUVÊA, M. M. Aprendizagem por Reforço Clássica e Conexionista: análise de aplicações. Anais do EATI - Encontro Anual de Tecnologia da Informação. Instituto Politécnico – Pontifícia Universidade Católica de Minas Gerais, p. 299-302, 2015.

CABRAL, D. Aprendizado por reforço – Um conto de Natal. (2018). Disponível em: <https://www.deviante.com.br/noticias/aprendizado-por-reforco-um-conto-denatal/> Acesso em: 14 set. 2020.

Obrigada!

hulianeufrn@gmail.com