Reinforcement Learning

João Pedro Dias[1]

Universidade de Évora, Évora, Portugal
m42055@alunos.uevora.pt

Resumo A inteligência artificial é um tema que tem vindo a ser cada vez mais impulsionado por diversas áreas, na tentativa de habilitar as máquinas a desempenharem tarefas de forma independente. O aumento da procura por soluções nesta área, pode também ser explicado pelo facto de a maior barreira a nível da aprendizagem automática, atualmente, consistir na programação de software ao invés do hardware das máquinas em si. No presente artigo, procede-se a uma abordagem sobre uma das técnicas de aprendizagem automática, denominada por aprendizagem por reforço ou *reinforcement learning*, que tem inovado diversos aspectos no âmbito do conhecimento artificial. Como tal, pretende-se explorar a maneira como se pode utilizar esta tecnologia, bem como evidenciar as suas vantagens e desvantagens.

Keywords: *Reinforcement Learning* · Aprendizagem · Inteligência Artificial · Agente · Ambiente

1 Introdução

Desde resultados surpreendentes no mundo dos jogos de vídeo até aos avanços impressionantes no mundo da robótica, é claro o aumento do interesse no campo de *reinforcement learning*. Comparado com outros métodos de aprendizagem automática, este tipo de aprendizagem por reforço veio trazer novas vantagens ao mundo da inteligência artificial como o facto de não necessitar de dataset de treino e de não ficar limitado a nível de conhecimento que o agente pode adquirir. Nesta perspectiva, este artigo aprofundará estas vantagens ao mesmo tempo que identifica desvantagens, não sem antes introduzir o conceito de *reinforcement learning* e a maneira como pode ser aplicado.

Em primeiro lugar, o foco será responder à pergunta sobre o que é *reinforcement learning*. Tentar desmistificar o conceito é muito importante nesta fase introdutória, para mais tarde se compreender os possíveis usos deste tipo de aprendizagem automática. Nesta fase do artigo, serão também apresentados os requisitos básicos para a formação de um sistema de aprendizagem por reforço, que permita a aprendizagem por parte do agente sobre um problema em estudo.

Após a introdução do conceito central do artigo, será explorado o contexto histórico do mesmo, com o intuito de perceber a relação com diferentes técnicas, como programação dinâmica e o processo de tentativa e erro. Será possível verificar a junção de duas temáticas, previamente independentes entre si, para formar o campo de *reinforcement learning* contemporâneo [1].

Enquadrado ainda no necessário para criar um sistema de reinforcement, será detalhado também neste artigo cada um dos elementos que o compõe. O propósito desta parte do artigo será caracterizar alguns termos, como por exemplo o significado de agente e quais as suas funções juntamente com as do ambiente. Para além destes dois conceitos tocar-se-á no significado de política, recompensa, função de valor e modelo do ambiente.

Na parte final do artigo, além de se discutir as vantagens e desvantagens do processo de aprendizagem automática, será ainda mencionado a estrutura base dos diferentes tipos de algoritmos disponíveis neste campo, para resolver problemas de aprendizagem. Dentro destes algoritmos será possível identificar algumas diferenças, havendo a possibilidade de fazer distinções sobre quais as situações mais indicadas para cada um.

2 O que é Reinforcement Learning?

O conceito de *reinforcement learning*, ou aprendizagem por reforço, pode ser definido como uma abordagem computacional de aprendizagem baseada em interações entre um agente e o ambiente a seu redor. Este método de aprendizagem é impulsionado pelas recompensas ganhas pelo agente ao efetuar cada uma das interacções com o ambiente a seu redor, tomando assim conhecimento, de forma independente, sobre quais os comportamentos mais benéficos para atingir um objectivo inicialmente estabelecido.

Ao contrário de outros modelos de aprendizagem automática, como a aprendizagem supervisionada e aprendizagem não supervisionada, em *reinforcement learning*, o agente não possuí nenhum conhecimento inicial proveniente de uma entidade externa [1]. Esta característica deve-se ao alto nível de interactividade que o problema a resolver contém, tornando-se extremamente complexo e pouco prático gerar exemplos de comportamentos desejáveis.

Dado que numa fase inicial do problema o agente não tem conhecimento sobre qual o melhor comportamento a desempenhar, deverá em primeira instância explorar o ambiente à sua volta para adquirir algum conhecimento [1]. Este processo tem como base a técnica de tentativa e erro, e poderá originar maus resultados, pois muitas destas tentativas de interações terão como consequência recompensas negativas. Por outro lado, o agente irá, nesta fase, adquirir informação, através do mapeamento entre ações e respetivas recompensas, que o guiará em ações futuras. No entanto, parte deste mapeamento não será direto, visto que algumas ações poderão desencadear recompensas imediatas negativas mas futuramente positivas, tornando o processo mais complexo e de longa duração.

3 Contexto Histórico

O conceito de aprendizagem por reforço como o conhecemos hoje teve origem em finais dos anos 80, através da fusão de duas temáticas distintas e independentes entre si [1]. A primeira dizia respeito ao processo de aprendizagem por tentativa e erro, e desde cedo esteve relacionada com a psicologia da aprendizagem animal.

O segundo tema, por outro lado, em nada estava relacionado com o processo de aprendizagem, tratando apenas de problemas de controlo óptimo e as possíveis soluções usando técnicas de programação dinâmica.

A primeira temática é toda ela baseada na ideia de aprender através do método de tentativa e erro. Este principio foi pela primeira vez descrito em 1911 por Edward Thorndike, que defendia que a partir de um conjunto de respostas sobre a mesma situação, aquelas que eram acompanhadas de um resultado satisfatório, para um animal, eram aquelas que teriam mais probabilidade de voltar a acontecer [6].

Com o surgimento dos primeiros vestígios de inteligência artificial, aparece também as primeiras aplicações, a nível computacional, de técnicas de tentativa e erro por parte de Alan Turin [1]. Foi especialmente através da construção de máquinas electromecânicas que se reproduziu este tipo de comportamento, dando aso à possibilidade de programação de software com o intuito de replicar tipos de aprendizagem, com base neste tópico.

É então no âmbito da programação de software que se foca a segunda temática a dar origem a *reinforcement learning*, mais concretamente programação dinâmica. Este tipo de programação continua a ser o melhor método para resolver problemas do âmbito de controlo óptimo, no entanto, segundo Bellman, sofre da maldição de dimensionalidade [1]. Bellman justifica que à medida que o número de possíveis estados de um problema aumenta, a necessidade de recursos computacionais aumenta também, reduzindo a eficiência do algoritmo.

Durante muitos anos a utilização de programação dinâmica esteve distanciada da aprendizagem automática, em particular, por se pensar que o facto de recorrer variadas vezes a computações realizadas no passado não tinha qualquer propósito para um processo de aprendizagem que é direccionado para o futuro. No entanto com o passar do tempo esta relação foi formada através de diversos trabalhos na área, como o tratamento de aprendizagem por reforço utilizando cadeias de Markov ou a combinação de programação dinâmica com redes neuronais artificiais.

Estes duas temáticas unem-se mais tarde em torno de uma terceira originando o campo de *reinforcement learning*, como o conhecemos hoje. Esta terceira temática, denominada de aprendizagem de diferenças temporais, consiste em diferentes métodos de aprendizagem guiados por diferentes estimativas sucessivas da mesma quantidade [1].

Atualmente, a investigação na temática da aprendizagem por reforço tem crescido exponencialmente, derivado, principalmente, ao seu foco na área da inteligência artificial [7]. Mas também é possível verificar o crescimento ligado a outras áreas como a psicologia e a optimização matemática, graças à sua vertente de aprendizagem comportamental. Já no ramo da inteligência artificial, é possível encontrar diversas aplicações de *reinforcement learning*, quer no âmbito da robótica, dos jogos de vídeo ou até mesmo ao nível de redes de computadores.

4 Sistema Reinforcement Learning

Existem diversos elementos, que apesar de independentes do contexto de um problema de aprendizagem, são fundamentais para formar um sistema de aprendizagem por reforço. Os mais importantes, e que devem ser caracterizados em primeira instância, são o agente e o ambiente. Mas para além destes dois elementos, também é necessário identificar uma politica de aprendizagem, uma função recompensa, uma função de valor e, em casos específicos, um modelo do próprio ambiente (fig.1).

Nesta secção tentar-se-á explorar com maior detalhe cada um destes conceitos, de modo a que no futuro, a tarefa de perceber se a técnica de aprendizagem por reforço é a mais adequada a um certo problema, seja facilitada.

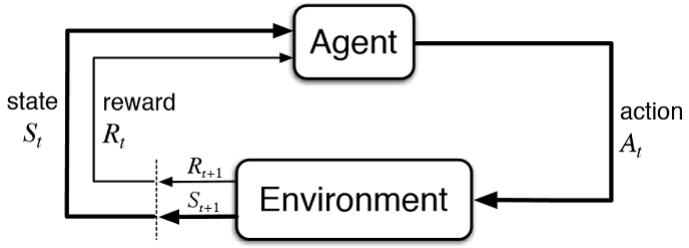

Figura 1. Representação de um sistema de *reinforcement learning*

4.1 Agente

O agente é considerado o papel central de um sistema deste tipo, pois é a seu redor que estará envolto todo o processo de aprendizagem. Irá ser o agente que interagirá com o ambiente através de um conjunto de ações, que lhe será disponibilizado à partida. Estas ações resultarão em recompensas (positivas ou negativas) que o agente recolherá com o intuito de melhorar a sua tomada de decisões [2]. Numa visão mais prática o agente será o componente de software que irá tentar resolver o problema da maneira mais eficaz [2]. Para isso, é também importante que este agente tenha um objectivo definido, para se mover durante todo o processo.

Existem três aspectos essenciais que devem ser garantidos por parte do agente de modo a ser viável a utilização de aprendizagem por reforço [1]. É necessária a existência de um objectivo sobre o qual o agente se possa guiar, sendo por isso, a definição deste objectivo o primeiro passo num sistema deste género. Para além deste aspeto o agente também terá de tomar as suas decisões com base nas percepções recolhidas através do ambiente, tornando-se obrigatório que o agente consiga sentir o estado atual do ambiente à sua volta. Por último, é necessário que o agente consiga alterar o estado do ambiente através das suas interações.

4.2 Ambiente

O ambiente, como o nome indica, é tudo o que está ao redor do agente, e onde este vai executar todas as suas ações. Estas ações irão, por sua vez, alterar o estado do ambiente, que corresponde a um conjunto de informação contidas no próprio ambiente que podem, ou não, ser observadas pelo agente. Estes estados podem ainda ser representados como nós numa cadeia de Markov, sendo os arcos da cadeia as possíveis interações dentro de cada estado. Como principal função, o ambiente terá de transmitir sensações ao agente, que influenciarão a sua tomada de decisões, e terá também de disponibilizar as recompensas, sejam elas positivas ou negativas, após cada ação executada [3].

4.3 Politica

A politica de aprendizagem é um aspecto também fundamental num sistema de aprendizagem por reforço, uma vez que é o que define a maneira do agente se comportar [1]. De certo modo, esta politica pode ser vista como um conjunto de regras que determina as ações que podem ser realizadas pelo agente, consoante o estado do ambiente. É por isso um elemento essencial para compreender o comportamento do agente, tornando-se importante definir uma boa politica para otimizar ao máximo o seu desempenho [2].

4.4 Recompensa

O conceito de recompensa, previamente mencionado, é algo que o agente vai receber como consequência das suas ações, neste tipo de sistemas. Pode ser positivo ou negativo, mas deve-se ter sempre em conta que o objectivo do agente será maximizar a recompensa ao longo de todo o problema [1]. Todas as possíveis recompensas ficarão definidas através de uma função recompensa, que deve ficar bem estruturada.

4.5 Função de valor

Para alem da função recompensa, é necessário estabelecer uma função de valor de modo a guiar o agente na melhor direção a longo prazo. Ou seja, podemos considerar um valor de um estado do problema como o máximo de recompensas que o agente pode adquirir partindo daquele estado em especifico. Esta informação torna-se importante para os casos em que um estado resulta sempre numa recompensa instantânea baixa, mas contem um valor alto porque é seguido de estados que oferecem recompensas maiores [4]. Assim o agente ganha visibilidade sob o quão positivo é estar num estado especifico e como fazer uma ação o vai influenciar no futuro.

4.6 Modelo do ambiente

O último elemento que podemos encontrar num sistema deste género, é o modelo do próprio ambiente. Este modelo é benéfico porque replica o ambiente onde o agente será inserido, oferecendo a oportunidade de planear ações. Com esta abordagem, o agente será submetido ao problema real já com um conhecimento inicial, reduzindo a necessidade de haver uma exploração tão extensa numa fase introdutória [1].

5 Exemplo de Reinforcement Learning

Um exemplo de um sistema onde pode ser aplicado aprendizagem reforçada é o problema do pêndulo invertido. Este problema consiste no equilíbrio de uma vara na vertical cuja base está presa a um carrinho amovível. Este carrinho tem como opções a movimentação na horizontal para atingir o objectivo principal de equilibrar a vara (fig.2).

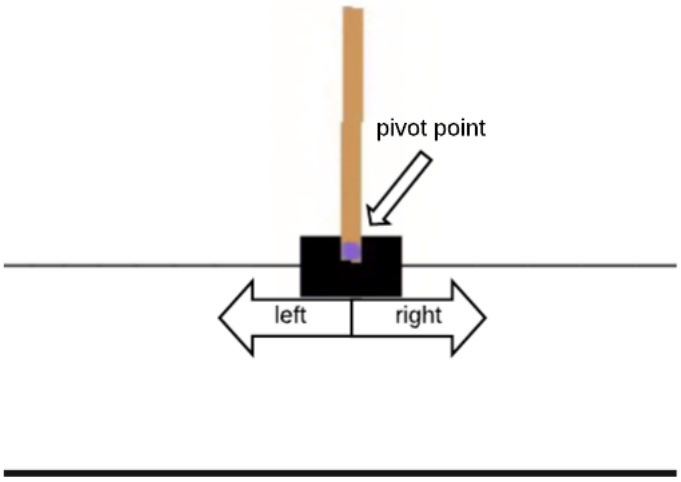

Figura 2. Representação do problema do pêndulo invertido

Neste sistema o agente que tentará resolver o problema será o carrinho, e terá como possíveis ações o deslocamento para a direita e para a esquerda. O ambiente será o conjunto de todos os elementos que formam o problema, neste caso o carrinho, a vara e o espaço de deslocamento, enquanto que os estados do ambiente serão a posição em que a vara se encontra em cada intervalo de tempo.

Sabendo que o objetivo do problema é aguentar o maior intervalo de tempo com a vara na vertical, é possível perceber que o carrinho irá receber uma recompensa sempre que conseguir manter a vara num intervalo angular aceitável em relação ao eixo vertical. Logo, a politica será a melhor estratégia que irá permitir ao agente não deixar a vara cair. Podemos ainda identificar uma função de valor que identifica os estados onde a barra faz um menor ângulo com a vertical como os mais favoráveis.

É por isso um problema simples, mas que mostra a potencialidade da implementação de aprendizagem por reforço, especialmente na área da robótica. Este comportamento, de equilibrar a barra, pode ser substituído por muitos outros, mantendo o mesmo tipo de sistema, e alcançando assim um tipo de inteligência artificial.

6 Algoritmos Reinforcement Learning

Num problema de *reinforcement learning* é habitual a utilização de Cadeias de Markov para a representação prática do problema. Numa Cadeia de Markov, cada nó pode ser representativo de um estado do ambiente, enquanto que cada arco corresponde a uma ação disponível para o agente. Por último a probabilidade de cada ação acontecer também é representada, sobre cada arco.

Com isto em mente é possível distinguir dois casos relativos à representação utilizando Cadeias de Markov. Caso todas as probabilidades de transição de estado e as possíveis recompensas forem conhecidas à partida, será viável construir uma cadeia de Markov completa. E através de uma cadeia completa, resolver o problema de *reinforcement learning* recorrendo a algoritmos de programação dinâmica, como o algoritmo de iteração de politica ou de iteração de valor [9].

No entanto, na grande maioria dos problemas de aprendizagem por reforço, não é conhecida toda a dinâmica, pelo que é impossível a construção de uma cadeia de Markov completa logo na fase inicial do problema. Por isso, é preciso usar a interação do agente com o ambiente para, através da aprendizagem adquirida, construir uma representação do modelo ou uma politica de decisões a tomar. Em ambos os casos a interação resultará em recompensas e estimativas para decisões futuras.

Para os casos em que o agente não sabe como o ambiente vai reagir em resposta ás suas interações, é necessário encontrar uma boa política para a tomada de decisões, existindo para isso duas abordagens diferentes para alcançar esse propósito [9]. Caso o agente produza um modelo do ambiente, internamente, através do resultado das suas interações, estaremos perante a abordagem *model based*. Nesta abordagem, após o agente adquirir conhecimento suficiente para

modelar o ambiente, pode utilizar algoritmos de planeamento para melhorar a sua tomada de decisões. Se por outro lado o agente optar por não criar um modelo do ambiente, estaremos perante uma abordagem *model free*. Desta maneira, o agente não necessita de modelar o ambiente para encontrar uma boa política, calculando apenas a utilidade de cada estado em que se encontra.

Como principal diferença entre estes dois tipos de algoritmos temos o facto de em *model base* ser necessário armazenar toda a informação recolhida ao nível das transições de estado, o que se pode tornar pouco prático à medida que o número de estados e ações aumenta. Já em *model free*, não é necessário armazenar estas combinações, focando-se mais na técnica de tentativa e erro para encontrar a melhor política.

É possível encontrar diversos algoritmos de aprendizagem por reforço baseados no conceito *model free*. Q-learning é um desses algoritmos e um dos mais utilizados. Este algoritmo tenta encontrar a melhor politica de modo a maximizar o valor de cada estado, para o agente ter o melhor comportamento possível. O algoritmo começa por atribuir um valor default a todos os estados e à medida que o agente realiza ações, atualiza o valor de Q, ou mais concretamente o valor de cada estado. Este cálculo é baseado na equação de Belman. Como este algoritmo existem ainda outros algoritmos como o Monte Carlo e SARSA.

7 Vantagens

Existem inúmeras vantagens em usar esta abordagem para desenvolver sistemas computacionais inteligentes [5], mas a principal é ser a única que permite que o agente interaja com o ambiente diretamente, para atingir o objectivo para o qual foi desenhado. É por isso diferente dos outros tipos de aprendizagem já que o agente aprende de forma independente e dinâmica, sendo apenas guiado por políticas previamente definidas.

Quando estamos perante um problema para o qual não dispomos de dados de treino, surge mais uma das vantagem em usar a técnica de aprendizagem por reforço. Nestes casos, continua a ser possível desenvolver um sistema inteligente, dado que o agente aprende através da sua interação momentânea com o ambiente, não sendo necessário qualquer tipo de conhecimento prévio. A falta de casos de treino pode também ser justificada pela dificuldade que existe em replicar exemplos para um problema complexo, sendo por isso a única solução optar pela interação direta entre o agente e o ambiente do problema.

Por último existe ainda mais uma vantagem ao usar este tipo de aprendizagem, especialmente no que diz respeito ao mundo dos videojogos. Ao desenvolver um agente que consiga jogar um certo jogo de maneira inteligente, ao ser usada aprendizagem supervisionada, este agente nunca será melhor que o jogador que originou o dataset de treino, ficando assim com conhecimento limitado. Já no caso de *reinforcement learning*, o agente apenas depende de si próprio para explorar e aprender quais as melhores jogadas, tornando a sua capacidade de aprendizagem naquele jogo ilimitada.

8 Desafios

Apesar de possuir muitos pontos fortes, esta metodologia de aprendizagem automática contém também desafios que necessitam de ser ultrapassados, para atingir melhores resultados e também melhor performance. Um dos principais desafios surge na fase inicial do problema. Nesta fase é quase garantido que o agente irá ter um mau desempenho, e isto deve-se à ausência de conhecimento inicial que o agente tem sobre o ambiente ao seu redor, para tomar as decisões mais acertadas. Logo, haverá casos em que será necessário explorar ações com recompensas desconhecidas, que se poderão revelar negativas. É por isso necessário dar algum tempo ao sistema, até começarem a aparecer melhores resultados.

O desafio de tentar balancear o comportamento do agente entre o ato de exploração de ações desconhecidas e de ações dentro do seu conhecimento é talvez o problema mais debatido entre os peritos da área [8]. Este dilema deve ser abordado durante a implementação do sistema, para evitar comportamentos indesejáveis por parte do agente. Ao explorar demasiado o desconhecido, as recompensas tenderão a ser mais negativas, mas no caso do agente evitar ao máximo explorar o ambiente, poderá perder recompensas maiores a longo prazo. A situação ideal é por isso permitir sempre que possível ao agente explorar o desconhecido mas tentando nunca por em causa a sua performance [2].

Dado que, para grande parte dos problemas, o agente precisa de reforçar a sua base de dados de conhecimento através da execução de uma grande quantidade de ações, surge um novo desafio relacionado com a necessidade de muitos recursos computacionais. Esta característica, faz com que seja uma técnica de aprendizagem que precise de um poder computacional muito alargado para obter bons resultados, e isto é algo que não está ao alcance de todos os programadores. Por outro lado, esta necessidade de processar uma grande quantidade de dados faz com que seja uma boa solução para a industria dos videojogos, pois a atividade de aprendizagem por parte do agente pode continuar indefinidamente ao mesmo tempo que o jogo fica disponível para jogar.

Existem ainda mais desafios, tanto ou mais relevantes que os mencionados anteriormente, e que continuam a ser estudados. É o caso de, por exemplo, um simples jogo de xadrez onde o movimento de uma peça a meio do jogo pode ter grande impacto no resultado final, mas transmitir isto para um sistema de aprendizagem por reforço torna-se difícil [2].

9 Conclusão

Chegado ao fim do presente artigo e consequente estudo sobre a matéria de aprendizagem por reforço, ou *reinforcement learning*, não ficam quaisquer dúvidas sobre o avanço feito na área e também o potencial para aplicação desta abordagem em diferentes áreas de aprendizagem. Ao descrever o conceito, foi possível identificar características únicas que distinguem este tipo de aprendizagem automática de outros, como aprendizagem supervisionada, sendo o facto de ser um processo independente e interactivo, por parte do agente, o ponto em destaque do método em estudo.

Foi também possível explorar a componente mais prática de *reinforcement learning*, ao distinguir diferentes tipos de algoritmos, revelando assim a variedade de opções disponíveis para a tentativa de solucionar um problema de aprendizagem.

Em última análise, a escolha de aprendizagem por reforço como solução principal em diversas áreas foi reforçada através da enumeração de múltiplas vantagens, como a possibilidade de se desenvolver agentes com melhor performance que qualquer humano em certos jogos. No entanto, a identificação de algumas desvantagens, também ajudou a perceber que para alguns casos esta pode não ser a melhor alternativa, principalmente para problemas onde se ambiciona os melhores resultados numa fase inicial da iteração com o problema.

Referências

1. Sutton, R. and Barto, A., 2018. Reinforcement Learning: An Introduction. 2nd ed. Cambridge, Massachusetts: The MIT Press.
2. Lapan, M., 2020. Deep Reinforcement Learning Hands-On. 2nd ed.
3. Ponteves, H., 2019. AI Crash Course. 1st ed.
4. Wiering, M. and Otterlo, M., 2012. Reinforcement Learning. 1st ed. Heidelberg: Springer.
5. Pros And Cons Of Reinforcement Learning | Pythonista Planet. `https://www.pythonistaplanet.com/pros-and-cons-of-reinforcement-learning` Last accessed 13 June 2020
6. Law Of Effect. `https://www.oxfordreference.com/view/10.1093/oi/authority.20110803100054653` Last accessed 25 June 2020
7. Wiering, M., Otterlo, M.: Reinforcement learning. Springer, Heidelberg (2012).
8. What is the explore-exploit tradeoff? - Explanation and examples — Conceptually. `https://conceptually.org/concepts/explore-or-exploit` Last accessed 27 June 2020
9. A Structural Overview of Reinforcement Learning Algorithms. `https://towardsdatascience.com/an-overview-of-classic-reinforcement-learning-algorithms-part-1-f79c8b87e5af` Last accessed 28 June 2020