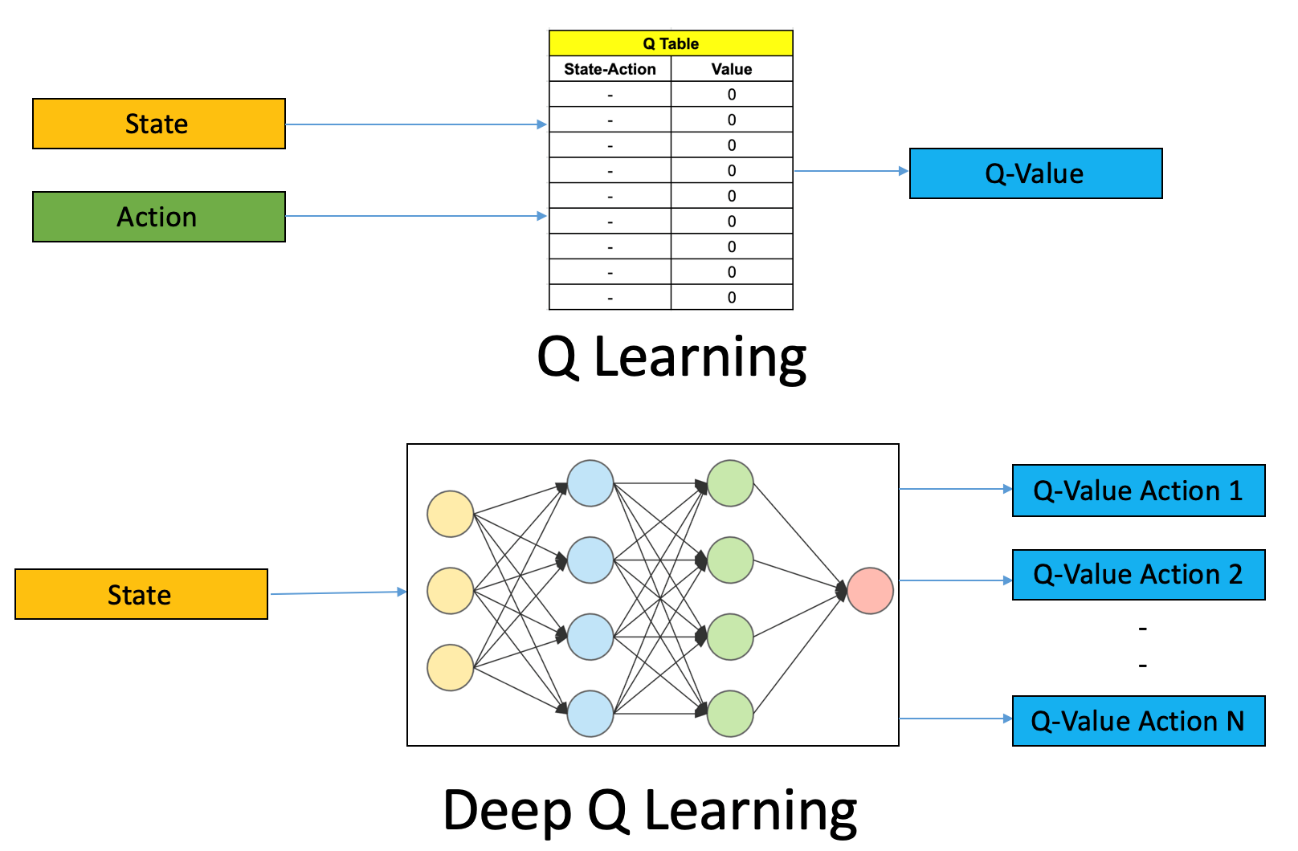
**Lucas Zillig Barros Oliveira**

**Deep Q-Learning**

Como vimos nas últimas aulas, o algoritmo Q-Learning produz uma tabela (Q-table), onde determinado agente usa para encontrar a melhor ação dado o estado. Um problema desse tipo de solução é a escalabilidade. Imagine um jogo, onde o estado é definido por coordenadas na tela. Em casos como esse, o algoritmo não é muito eficiente.

A partir desse problema, surge uma abordagem chamada Deep Q-Learning. Nela, ao invés de passarmos o estado e a ação para uma tabela, passamos o estado para uma rede neural que dá como output um valor (Q-value) para cada ação possível, sendo que quanto maior esse valor, melhor é a ação. Podemos comparar o funcionamento dos algoritmos observando a figura abaixo.

A “experiência”, ao invés de ser armazenada em uma tabela, é salva em memória. A função de perda é erro quadrado médio do Q-valor predito com o Q-valor “target”. Parece até um problema de regressão, porém não sabemos o real valor desse target.

Como já mencionado antes, a input layer de nossa rede neural será o estado. Nas hidden layers, criamos duas: uma para o Q-valor predito e outra para o Q-valor target. Por fim, a output layer retornará um valor para cada ação possível (ou seja, se houverem quatro ações, retornará quatro valores).

Por fim, outro ponto importante desse algoritmo é o “experience replay”, onde as experiências do agente são armazenadas, o que nos permite lidar com dois pontos:

1. Não esquecer experiências passadas: criamos uma espécie de buffer, de onde tiramos uma amostra (chamada de batch) para passar para a rede neural. Assim, experiências passadas realmente influenciam no comportamento do agente.
2. Correlação entre as experiências: ao selecionar uma amostra aleatória, evitamos que o agente obtenha algum tipo de viés. Por exemplo, se sempre usássemos as últimas experiência, algum efeito aleatório poderia ocorrer de maneira seguida, fazendo-o adotar aquilo como um padrão do ambiente. É um conceito até que parecido com exploration e explotation.

**Referências:**

https://s3-ap-south-1.amazonaws.com/av-blog-media/wp-content/uploads/2019/04/Screenshot-2019-04-16-at-5.46.01-PM.png

https://stats.stackexchange.com/questions/249355/how-exactly-to-compute-deep-q-learning-loss-function

https://www.freecodecamp.org/news/an-introduction-to-deep-q-learning-lets-play-doom-54d02d8017d8/

https://en.wikipedia.org/wiki/Q-learning#Deep\_Q-learning