Q-Learning exato

$$\text{difference} = \left[r + \gamma \max_{a'} Q(s', a')\right] - Q(s, a)$$

$$Q(s, a) \leftarrow Q(s, a) + \alpha\,[\text{difference}]$$

Ação
Estado
Próximo estado
Recompensa imediata

Ações

Estados

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
...	...	...	...

Ações

Estados

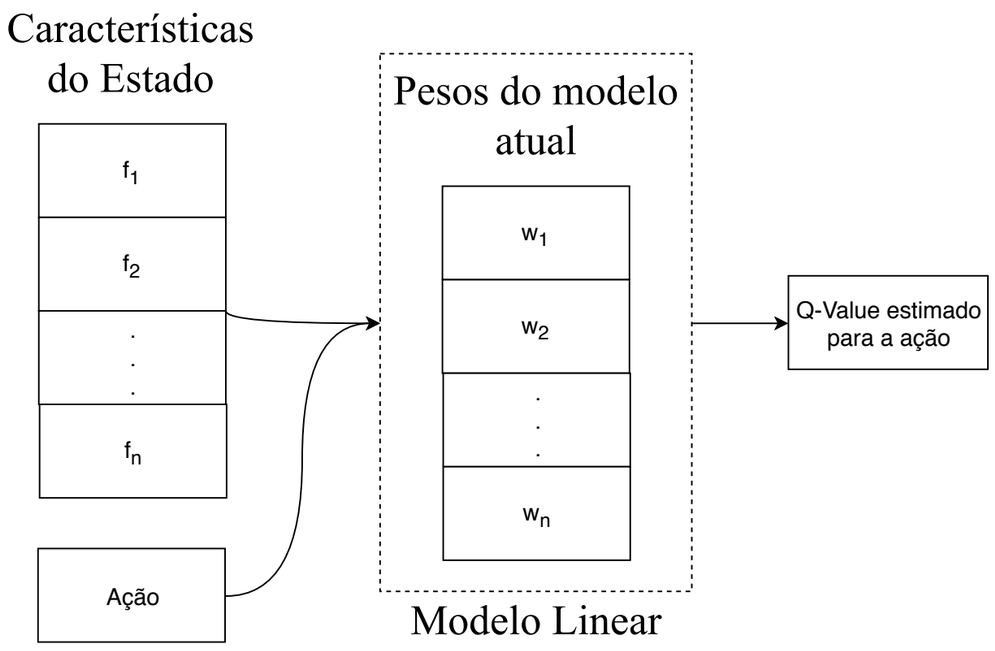
0	0	0	0
0	0	0	0
-0.4	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
...	...	...	...

Q-Learning aproximado: Versão Função Linear

$$Q(s, a) = w_1 f_1(s, a) + w_2 f_2(s, a) + \ldots + w_n f_n(s, a)$$

$$\text{difference} = \left[r + \gamma \max_{a'} Q(s', a')\right] - Q(s, a)$$

$$w_i \leftarrow w_i + \alpha\,[\text{difference}]\, f_i(s, a)$$



Q-Learning aproximado: Versão Multi-layer Perceptron

$$\text{Q-Value esperado} = R(s, a, s') + \gamma \max_{a'} Q(s', a')$$

Treinamento parcial da rede (relacionada a ação executada) utilizado o estado anterior e o Q-Value esperado