

WORKSHOP
APRENDIZADO POR REFORÇO COM

PONG



GRUPO
TURING

INTRODUÇÃO

GRUPO
TURING

O que é Machine Learning?

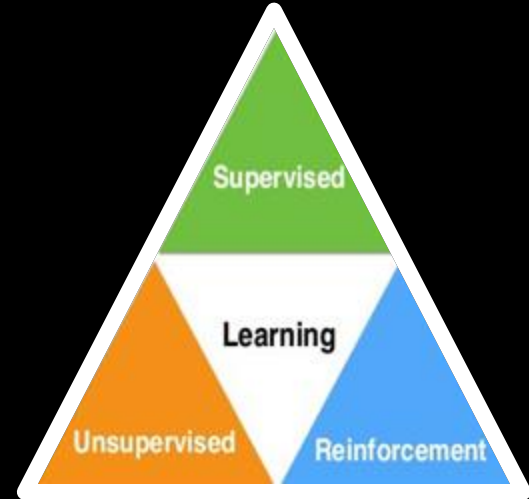
- Termo criado por Arthur Samuel em 1959:
- A área de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados



Tipos de Aprendizado

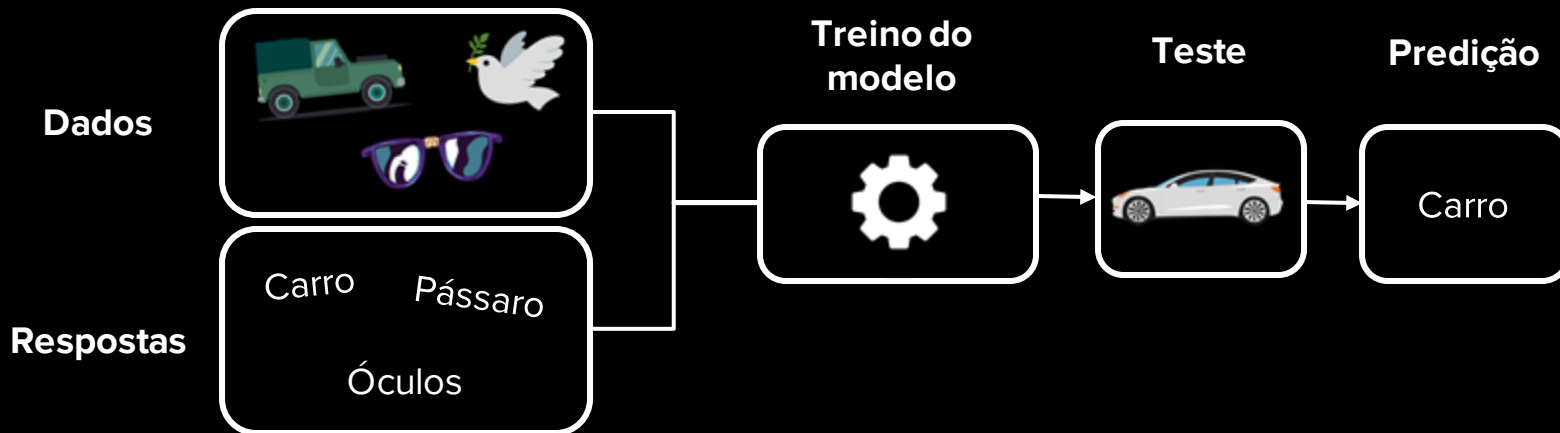
Como algo pode ser aprendido?

- **Aprendizado Supervisionado**
- **Aprendizado Não Supervisionado**
- **Aprendizado por Reforço**



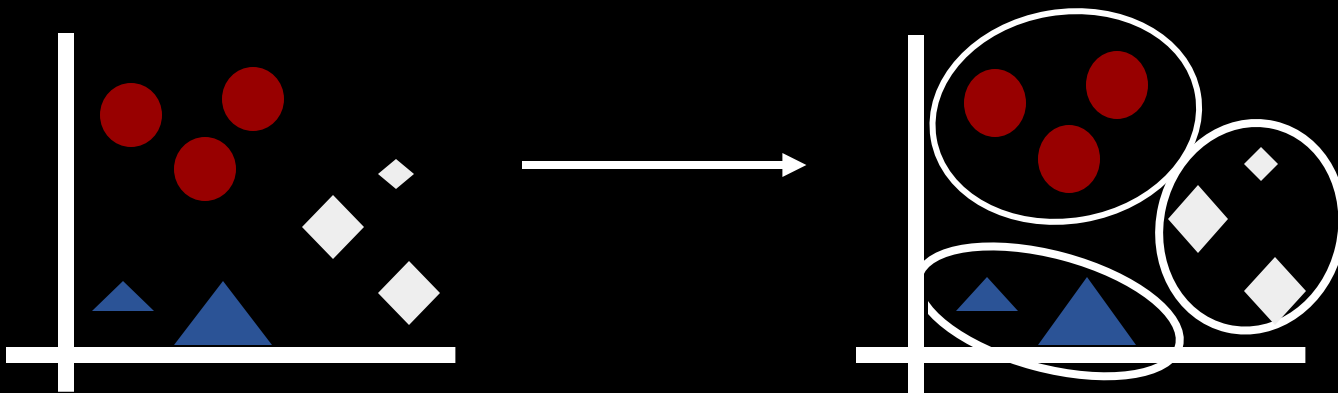
Aprendizado Supervisionado

- Aprendizado a partir de dados **já classificados**
- Tarefas como **classificação** e **regressão**



Aprendizado não Supervisionado

- Inferência a partir de dados **sem respostas**
- Tarefas de **segregação** e **associação**



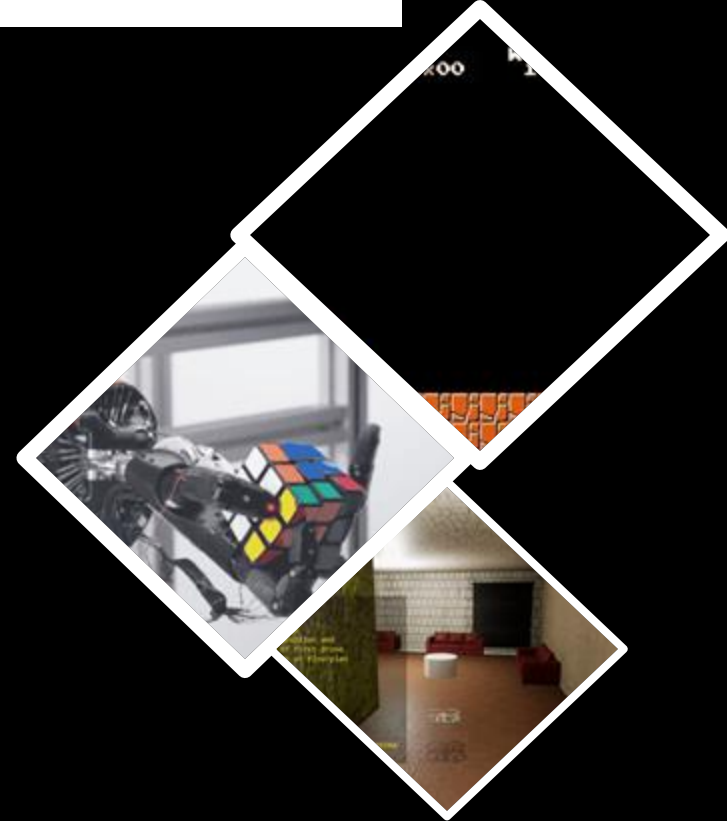
Aprendizado por Reforço

- Aprendizado por **tentativa** e **erro** (dados obtidos pela experiência)
- O modelo tenta aprender um comportamento para **maximizar** sua performance
- **Exemplo:** um cachorro que aprende um comportamento com base em se seu dono lhe dá uma bronca ou um petisco



Onde usamos Aprendizado por Reforço?

- **Robótica**
 - Controle de drones
 - Automação
- **Jogos**
- **Cuidados de Saúde**
 - Dosagem
- **Mercado Financeiro**
 - Previsão de ações
 - Transações



Teste o seu aprendizado

Você diria que as tarefas abaixo se enquadram em qual dos **3 tipos de aprendizado** mencionados?

- Dizer se uma foto é de um cachorro ou gato a partir de outras fotos
- Predizer o preço de uma casa
- Agrupar clientes de uma loja
- Maximizar o *score* de um jogo



CONCEITOS

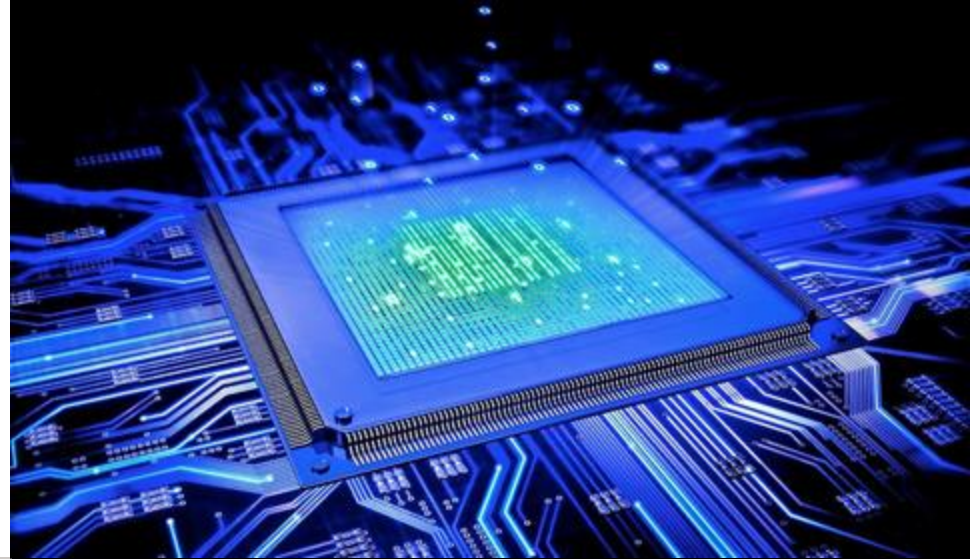


GRUPO
TURING

Agente

É o nosso *software*.

É a parte responsável por **tomar decisões**. E aprender as **melhores ações**.



O Agente não é exatamente um “personagem”, mas podemos pensar assim para entender alguns exemplos.

Agente

Podemos até entender o “**Mario**” como o **agente** do nosso jogo, mas o agente seria o *software* que comanda suas ações.



Analogamente, poderíamos pensar que o **jogador** é o “**agente**” no xadrez, e não o Peão.



Ambiente

Mundo com o qual o Agente interage.

É o **espaço** que representa nosso problema, transmitindo informações ao Agente.

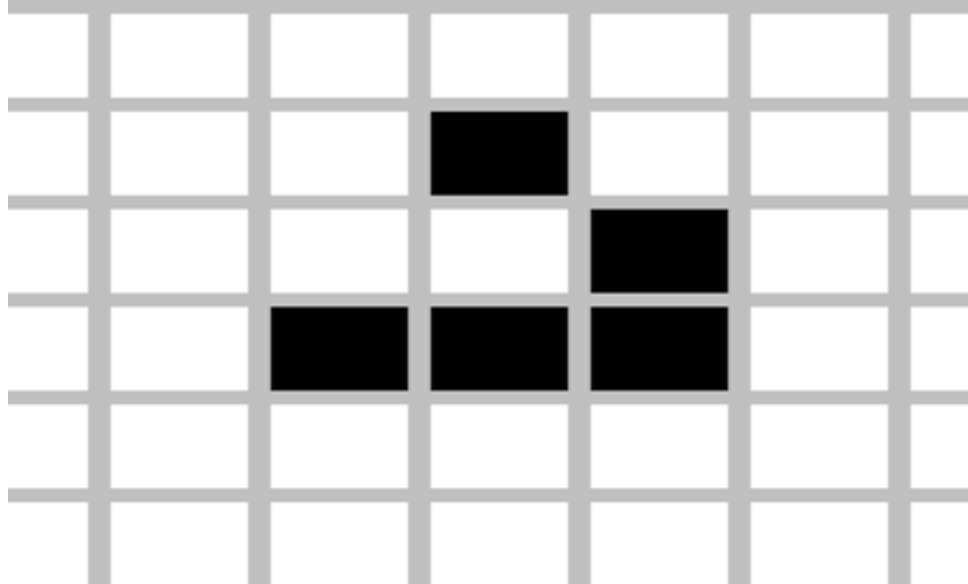
Vai além do espaço “físico”, também inclui personagens, sujeitos, seres...



Estado

É a descrição em um instante das condições do **Agente** e do **Ambiente**.

É a partir dele que o Agente deve tomar suas decisões.



A diferente distribuição das peças de xadrez configuram **estados** diferentes do jogo.

Ambientes Parcialmente Observáveis

Alguns ambientes não fornecem a informação completa do estado.



O agente pode deduzir o restante das informações com base na memória.

Ex: jogos de estratégia, carro autônomo.

Ação

Comando que o Agente **escolhe** para **interagir** com o ambiente.

No Xadrez, seria equivalente a um **movimento**.



Espaço de Ação

Conjunto de todas as ações possíveis.

No Xadrez, equivale ao conjunto de todos os **movimentos possíveis**.

Espaço de Ação

Discreto

Quantidade **finita** de ações

Simples de lidar

Ex: Controle (10 botões)



Contínuo

Intervalo com **infinitas** ações

Mais complexo

Ex: Velocidade (Entre 0 e 180 km/h)



Recompensa

A cada ação tomada, o Ambiente devolve um **feedback** ao Agente relatando a **efetividade** daquela ação.

Pode ser **positiva**, **negativa**, ou **nula**.

Ex: pontuação de um jogo.



Em casos em que criamos nosso próprio ambiente, devemos **modelar** nós mesmos as recompensas.

Se quisermos um time **agressivo** de futebol, por exemplo, podemos dar uma recompensa de **+2** para cada gol feito e **-1** para cada gol tomado.

Se quisermos **desincentivar** faltas, podemos penalizá-las com uma recompensa **negativa**.



PLAYER 1

00000500

HIGH SCORE

00130000

PLAYER 2

00000100

GENERIC VIDEO
GAME FONT 01

Retorno

HIGH SCORES

130000	WILLM
120000	GENE1
110000	FKING
100000	DJNIP
90000	FR007
80000	P2PNT
70000	APHOR
60000	FNTCL
50000	THLMC
40000	CMUNK

O objetivo do nosso
Agente é **maximizar** a soma
de todas as **recompensas**.

Essa soma de recompensas
a partir de um instante é
chamada **Retorno**.

Ou seja, se a **Recompensa**
era equivalente aos
Pontos de um jogo, o
Retorno é análogo ao
Score Total.

© 2006-2011 R.MEEK ELECTRONICS
INSERT COIN CREDIT 00

Retorno

O **Retorno** é obtido a partir da seguinte equação:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots$$

No cálculo do Retorno, somamos todas as recompensas multiplicadas por um **fator de desconto** (γ) entre 0 e 1.

Esse fator faz com que as recompensas mais para o futuro se tornem para vez menores, fazendo o **Retorno** convergir para um valor real.

- Um γ próximo de 1 significa que nosso Retorno leva muito em conta as **recompensas futuras**.
- Um γ próximo de 0 significa que levamos mais em conta **recompensas recentes**.



00046

1 £200

Política (π)

A política é o que guia as escolhas do nosso agente, fornece qual a próxima ação a ser tomada com base no estado atual.

Em jokenpô, jogar aleatoriamente pedra, papel ou tesoura seria uma política, assim como escolher uma sequência dos três.

O objetivo do nosso agente é encontrar a política ótima que escolhe a melhor ação para cada estado.

A melhor ação é aquela que nos leva ao maior retorno.



Valor (v)

O **Valor de um Estado** específico consiste no retorno esperado a partir daquele determinado estado.

$$V(s) = E[G_t | S_t = s]$$

$$V(s) = E[R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots | S_t = s]$$

Basicamente o valor que representa a recompensa total que costumamos receber após passar por aquele estado, ou seja, **quão bom é estar naquele estado**.

0.51 ▶	0.72 ▶	0.84 ▶	1.00
▲ 0.27		▲ 0.55	-1.00
▲ 0.00	0.22 ▶	▲ 0.37	◀ 0.13

Valor (v)

Com o **Valor** de um **Estado**, podemos escolher **Ações** que nos levem a **Estados** que tenham maior **Valor**.

Se o valor de um estado **S1** é maior que o valor de um estado **S2**, devemos tentar chegar em **S1**.

0.51 ▶	0.72 ▶	0.84 ▶	1.00
▲ 0.27		▲ 0.55	-1.00
▲ 0.00	0.22 ▶	▲ 0.37	◀ 0.13

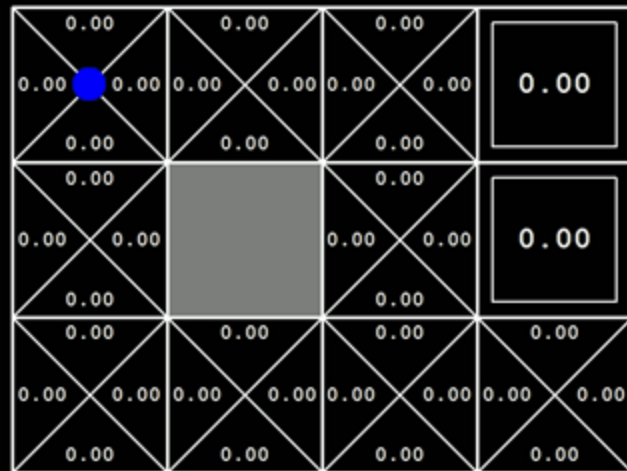
Valor-Ação (q)

O **Valor de uma Ação** consiste no retorno esperado a partir do momento em que se toma aquela ação.

$$\begin{aligned} q_*(s, a) &= E_{\pi^*} \left[G_t \mid S_t = s, A_t = a \right] \\ &= E_{\pi^*} \left[R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots \mid S_t = s, A_t = a \right] \end{aligned}$$

Dessa forma, o valor **q** de uma ação representa sua **qualidade**, ou quão bom é tomar aquela ação em um determinado estado.

O objetivo de muitos algoritmos de Aprendizado por Reforço é **estimar** os valores **q** de cada ação, para então escolher quais ações tomar escolhendo aquela de maior **q**.

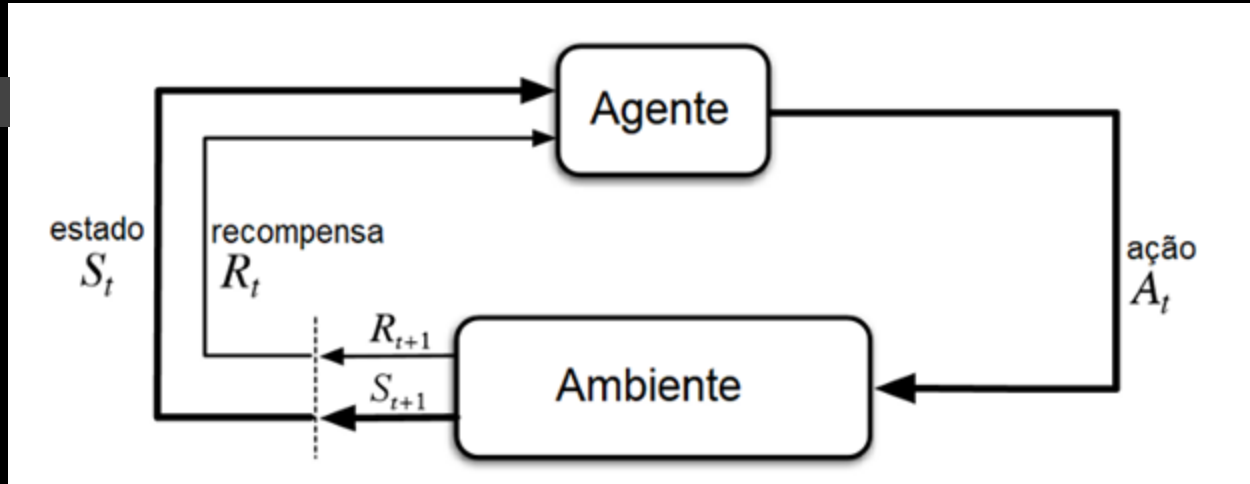


CURRENT Q-VALUES

Resumo

O **Agente** interage com o **Ambiente** por meio de uma **Ação** escolhida por uma **Política**, e recebe uma **Recompensa** e um novo **Estado** para escolher a próxima **Ação**.

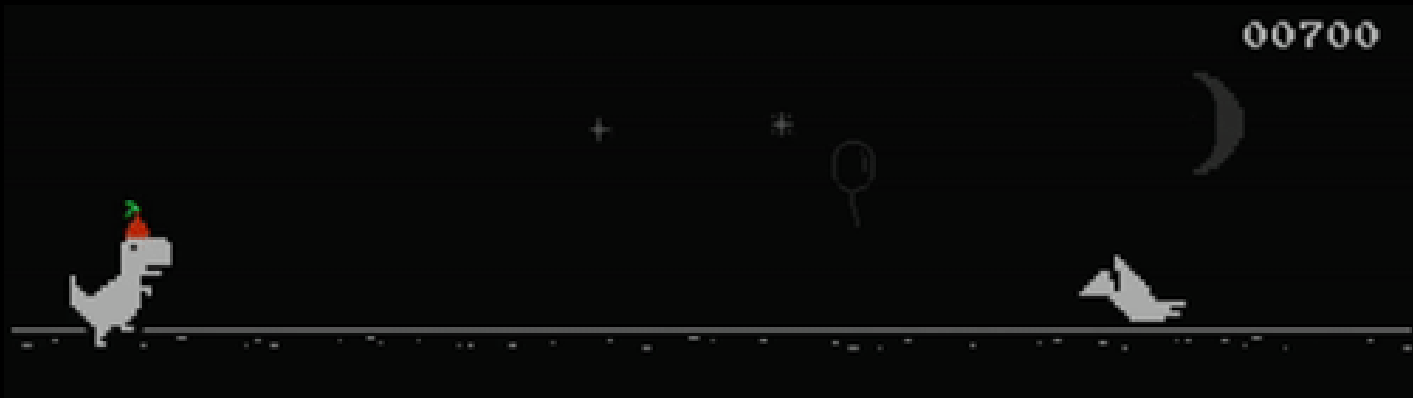
Escolher a ação
com maior Q



Teste o seu aprendizado

Dado o jogo ao lado, controlado por RL, como você descreveria...

- O Agente
- O Ambiente
- Uma ação
- Espaço de ação
- O estado



Pronto para fazer seu próprio agente?

