



Sobre o Curso

Semanas

- 1 Aula por semana
 - ~1h30min
 - Presença: Quem estiver na chamada
- 1 Laboratório por semana*
 - 2 semanas para entregar
- Slides e Labs estarão disponíveis no Github

Entregando Laboratórios

- Basta subir o código/notebook no github e enviar pra gente.
- De preferência use um único repositório para o curso todo.



Dúvidas e Discussões





Aprendizado por Reforço

AULA - 0

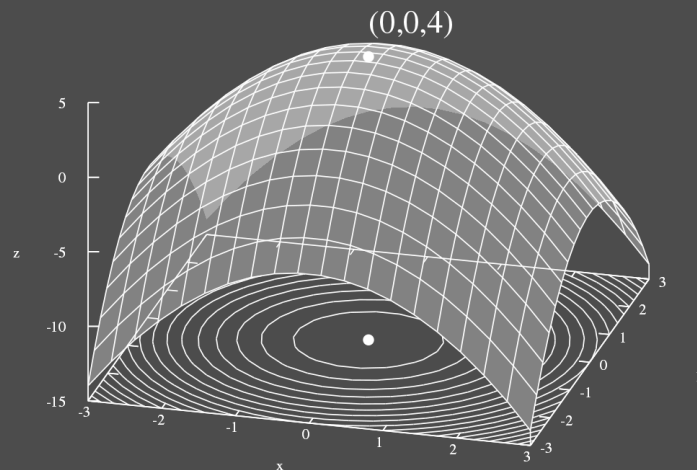
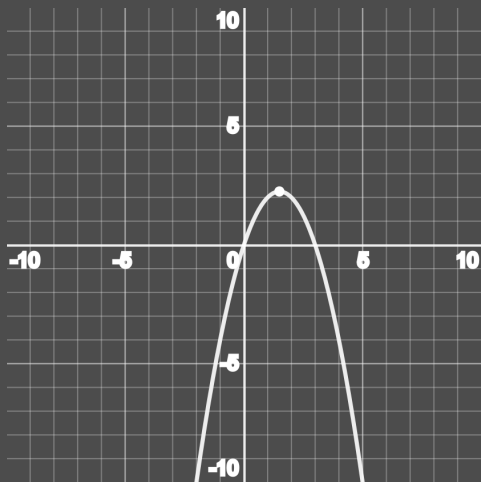
**Aprendizado de Máquina e
Imitation Learning**



Machine Learning

Otimização

- O que é otimizar?
 - Encontrar a melhor solução
- Maximizando ou Minimizando uma função

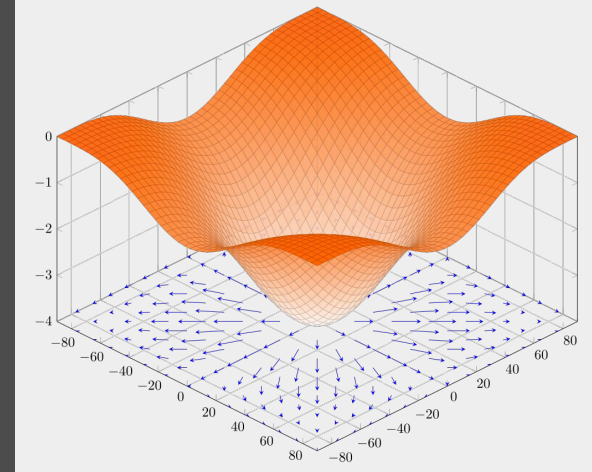
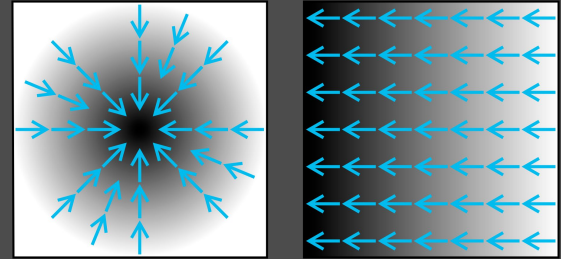


Gradiente

- Direção do maior aumento em uma função

$$\nabla f$$

$$\nabla f(x, y, z) = \frac{\partial f}{\partial x} \mathbf{i} + \frac{\partial f}{\partial y} \mathbf{j} + \frac{\partial f}{\partial z} \mathbf{k}$$



Tipos de Aprendizado

Supervisionado:

Aprender padrões a partir de exemplos

Não Supervisionado:

Encontrar padrões sem exemplos

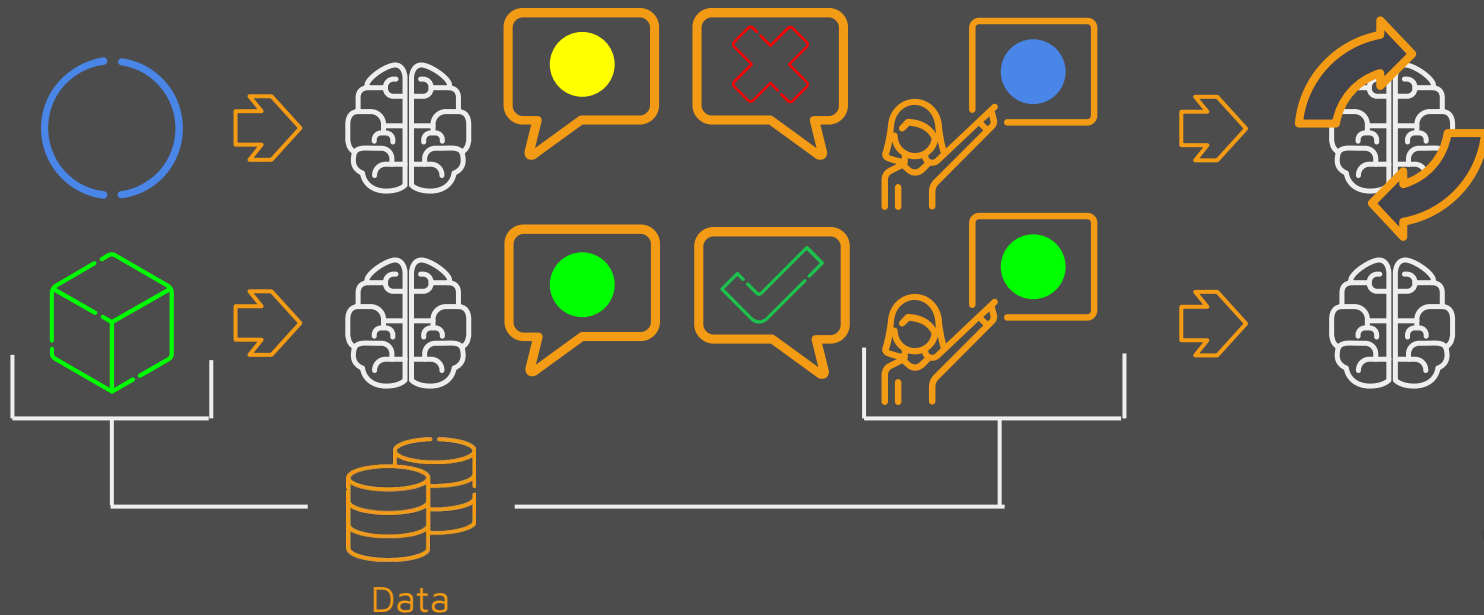


por Reforço:

Aprender a partir de interação e experiência

Aprendizado Supervisionado

- Com exemplos de comportamento correto



Depois do Treino

- Generalizar a partir de padrões “aprendidos”
- Reconhece os mesmos padrões em outras situações



O que está sendo otimizado?

- Função de Perda (*Loss Function*)
- Erro Quadrático Médio

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2.$$

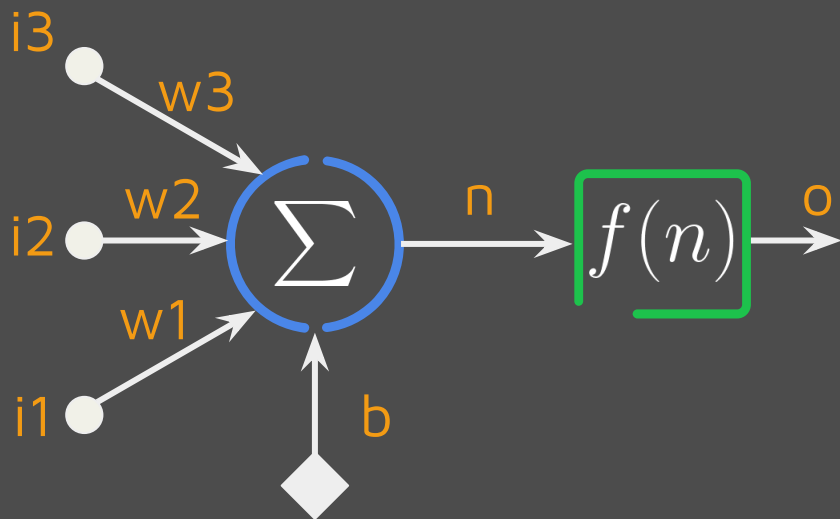
- Diferença entre a resposta correta e a resposta do modelo
- Minimização (contra o gradiente)



Redes Neurais Artificiais

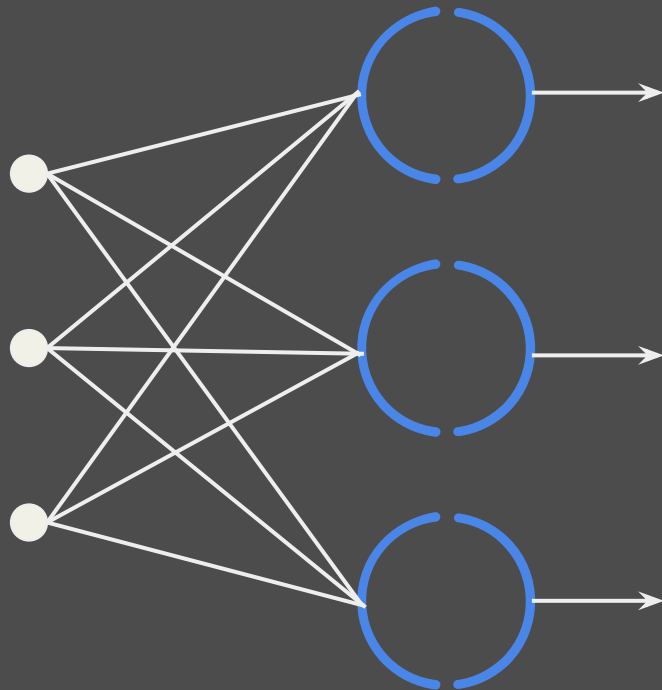
Neurônio Artificial

- Sinais de entrada são ponderados
- Somatória é passada por uma função
- Resultado é o sinal de saída



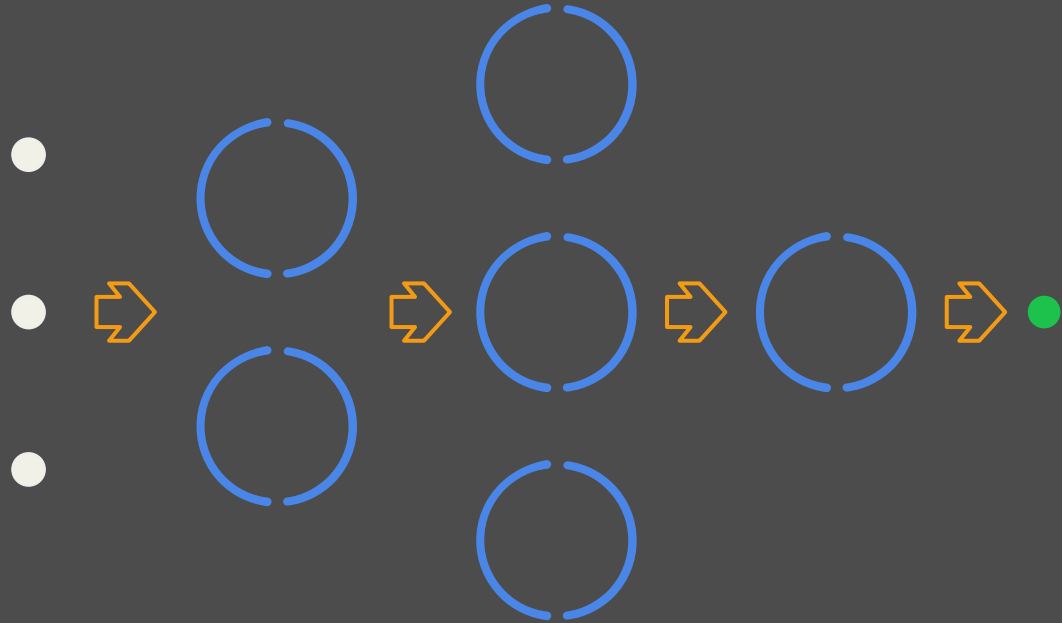
Camadas de Neurônios

- Vários neurônios juntos
- Entradas iguais
- Pesos diferentes
- Cada neurônio tem um sinal de saída



Rede Neural

- Camadas Sequenciais
- Entrada de um camada é a saída da anterior



Pesquisar em casa:

- Camadas Convolucionais
- Camadas Recorrentes



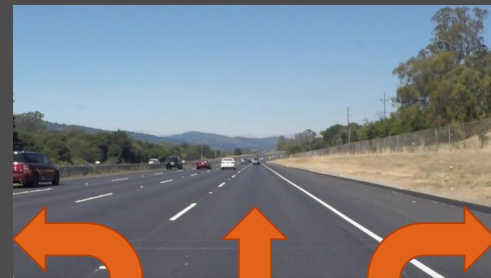
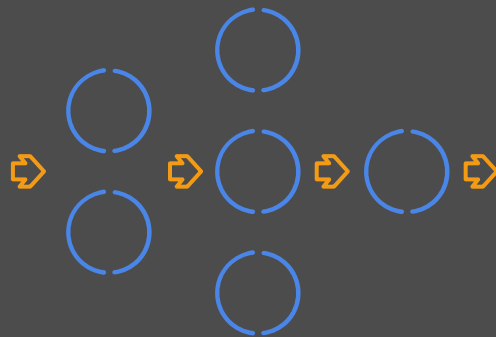
Imitation Learning

Aprendendo por Imitação

- Imagine que estamos tentando fazer um carro autônomo



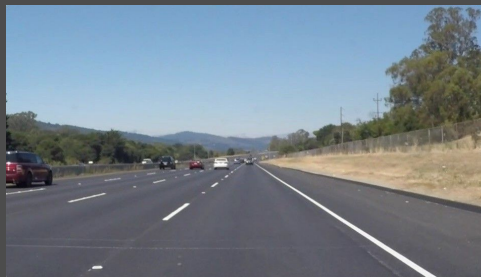
situação
observação



ação

Aprendendo por Imitação

- Podemos gravar uma pessoa dirigindo e guardar as situações e ações que ela toma



Behavior Cloning

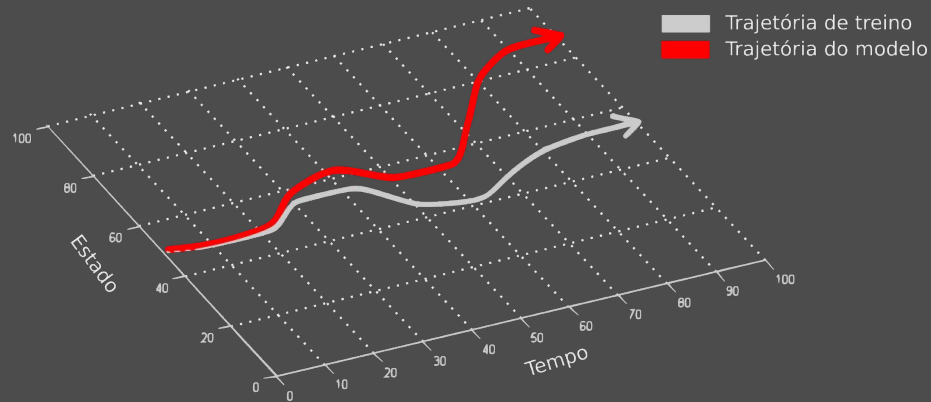
- Adquirir dataset de observações e ações do expert
- Treinar o modelo com aprendizado supervisionado, minimizando a função de perda

$$Data : \{(s_t, a_t^*), (s_{t+1}, a_{t+1}^*), (s_{t+2}, a_{t+2}^*) \dots\}$$

$$L = -\log(P(a^*|s))$$

Funciona?

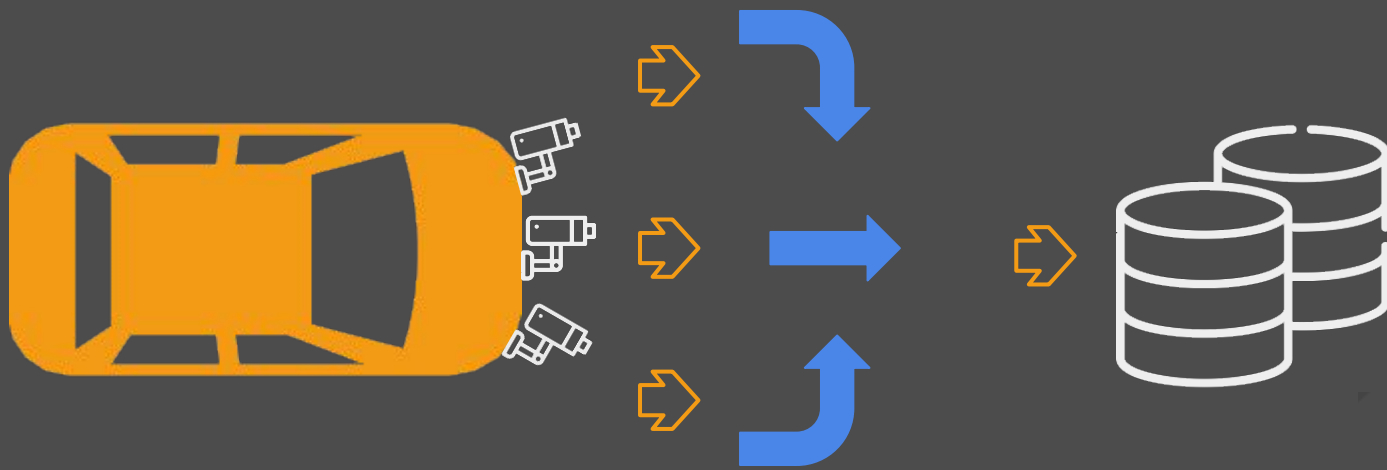
- Não tão fácil
- Dados mostram um bom comportamento
- Erros acumulam para situações muito diferentes



- É preciso ter REPRESENTATIVIDADE

NVIDIA

- Câmeras levemente laterais (extra)
- Ações com leves alterações p/ essas câmeras (mais dados)







Dagger here reacts dynamically to an untrained obstacle



PyTorch