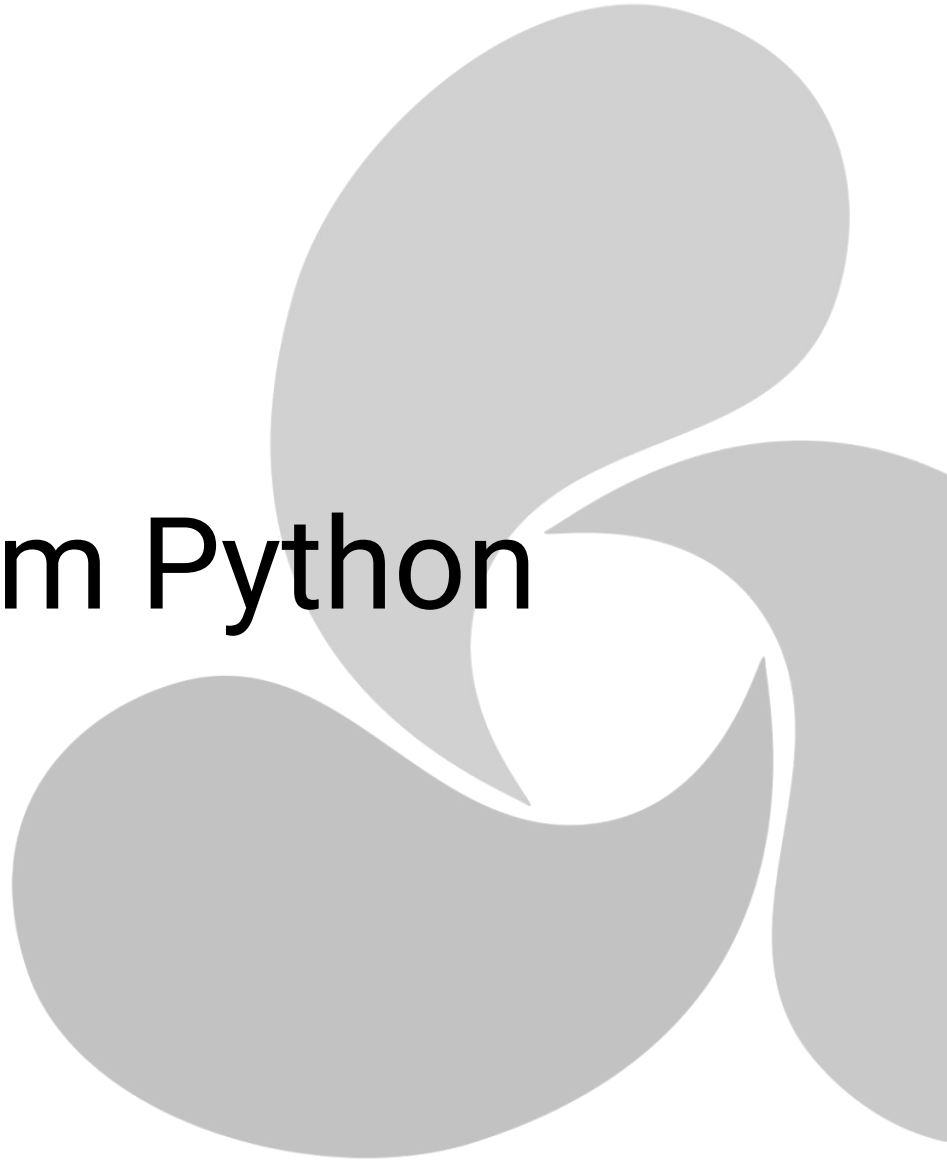
Machine Learning:
uma introdução prática em Python

Otacílio “Minho” Neto, minhotmog@gmail.com



Quem sou eu?

Uma pergunta difícil de se responder...

Otacílio “Minho” Bezerra

Olá! Eu sou o **“Minho”**, aluno de *Engenharia de Computação* e pesquisador em *Machine Learning*.

minhotmog@gmail.com

github.com/TioMinho

… também um amante de memes e cachorros (e de memes de cachorro).

Sobre o Minicurso

- O minicurso terá duração de 3 dias:
  - **1º Dia -** Introdução a Machine Learning e Regressão;
  - **2º Dia -** Regressão Polinomial e Classificação;
  - **3º Dia -** Redes Neurais Artificiais: SLP e MLP;

- Programação em Python 3.6 (Numpy+Pyplot);

- **+** Aprendizagem de Máquina
  - Data Science/Data Mining

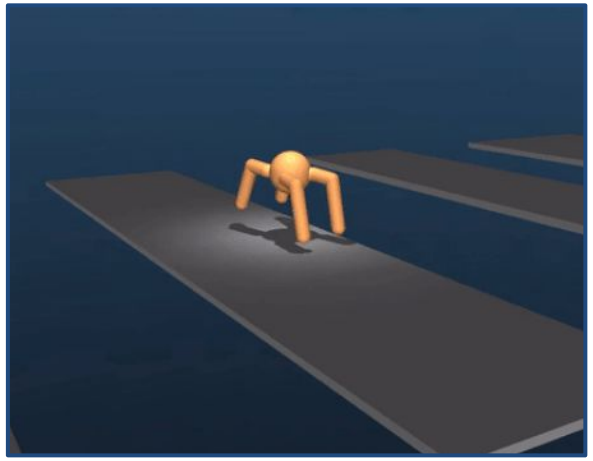
Por que estudar Machine Learning?

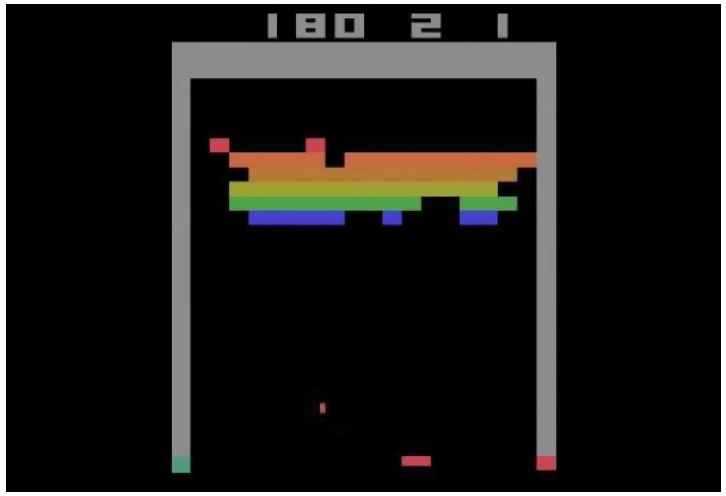
- **Ferramenta Poderosa -** as técnicas existentes já permitem automatizar até tarefas de altíssimo nível (e.g., *Sistemas de Recomendação*);

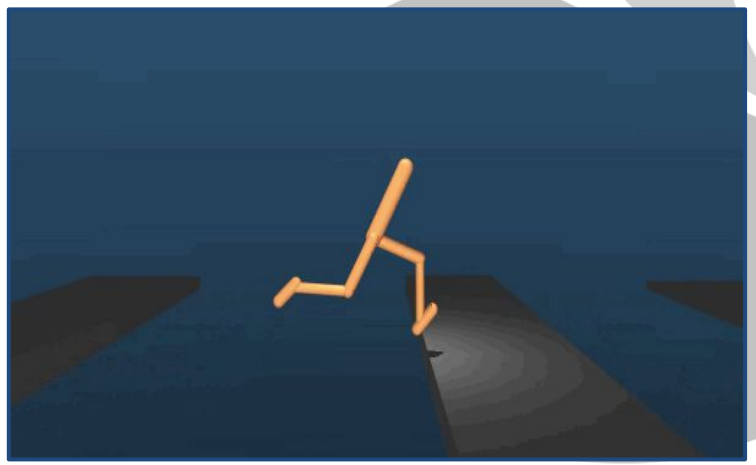
- **Está na moda! -** executivos da Google™ chegaram a afirmar que o futuro da Computação pertence à Inteligência Artificial por meio do Machine Learning;

- **Área em Crescimento -** precisamos de mais cientistas e engenheiros para desbravar essas terras ainda desconhecidas!

Deep Learning e a Google DeepMind







Introdução a Machine Learning

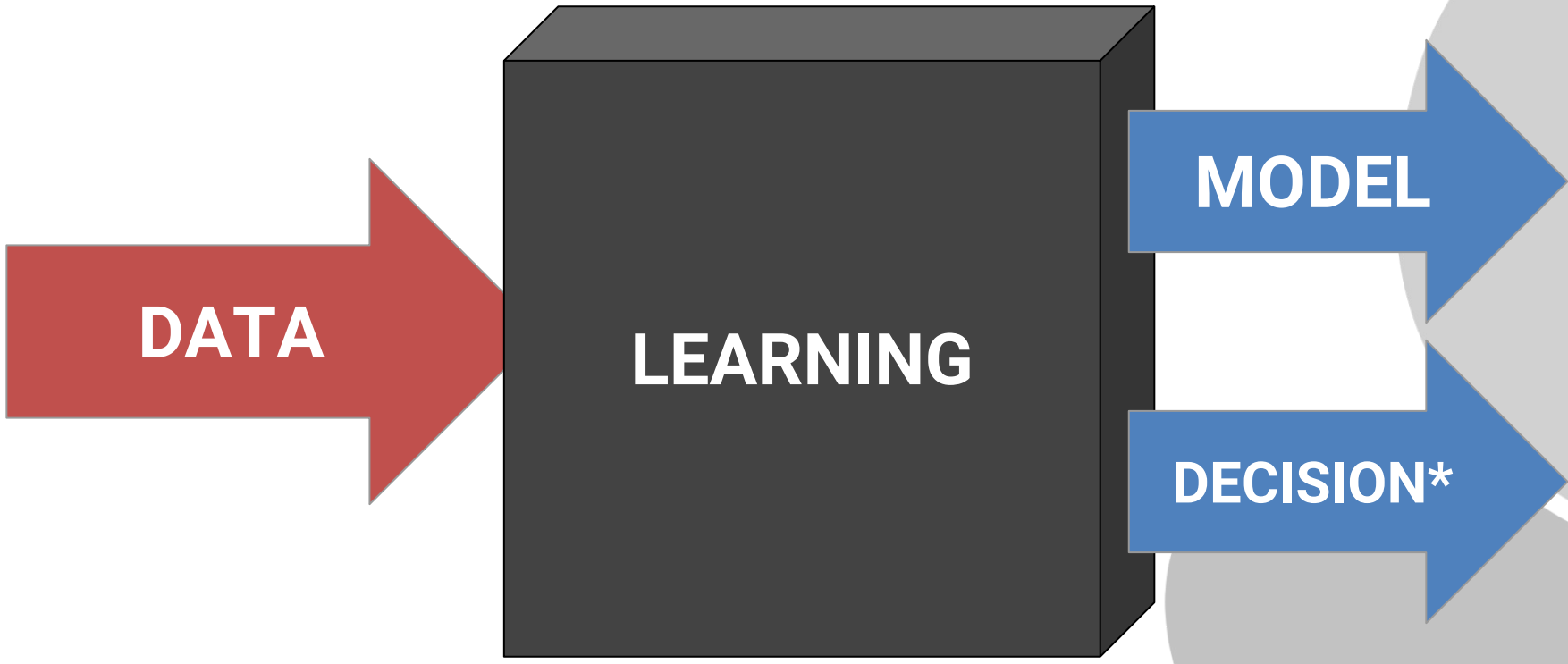
Compreendendo os conceitos iniciais dessa poderosa ciência.

O que é *Machine Learning*?

- **Segundo (Arthur Samuel, 1959):**

"gives computers the ability to learn without being explicitly programmed"

O que é *Machine Learning*?

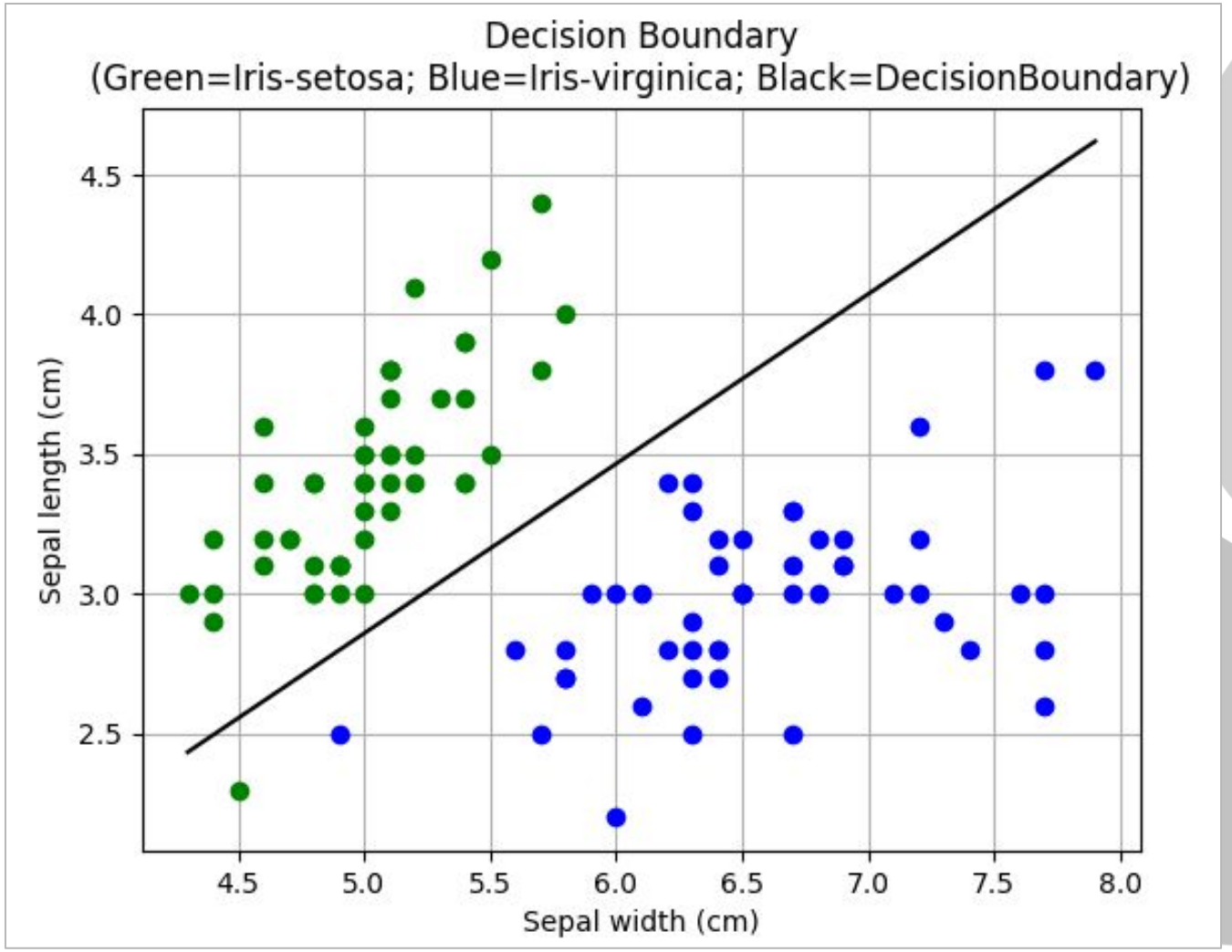


* Sistemas Híbridos

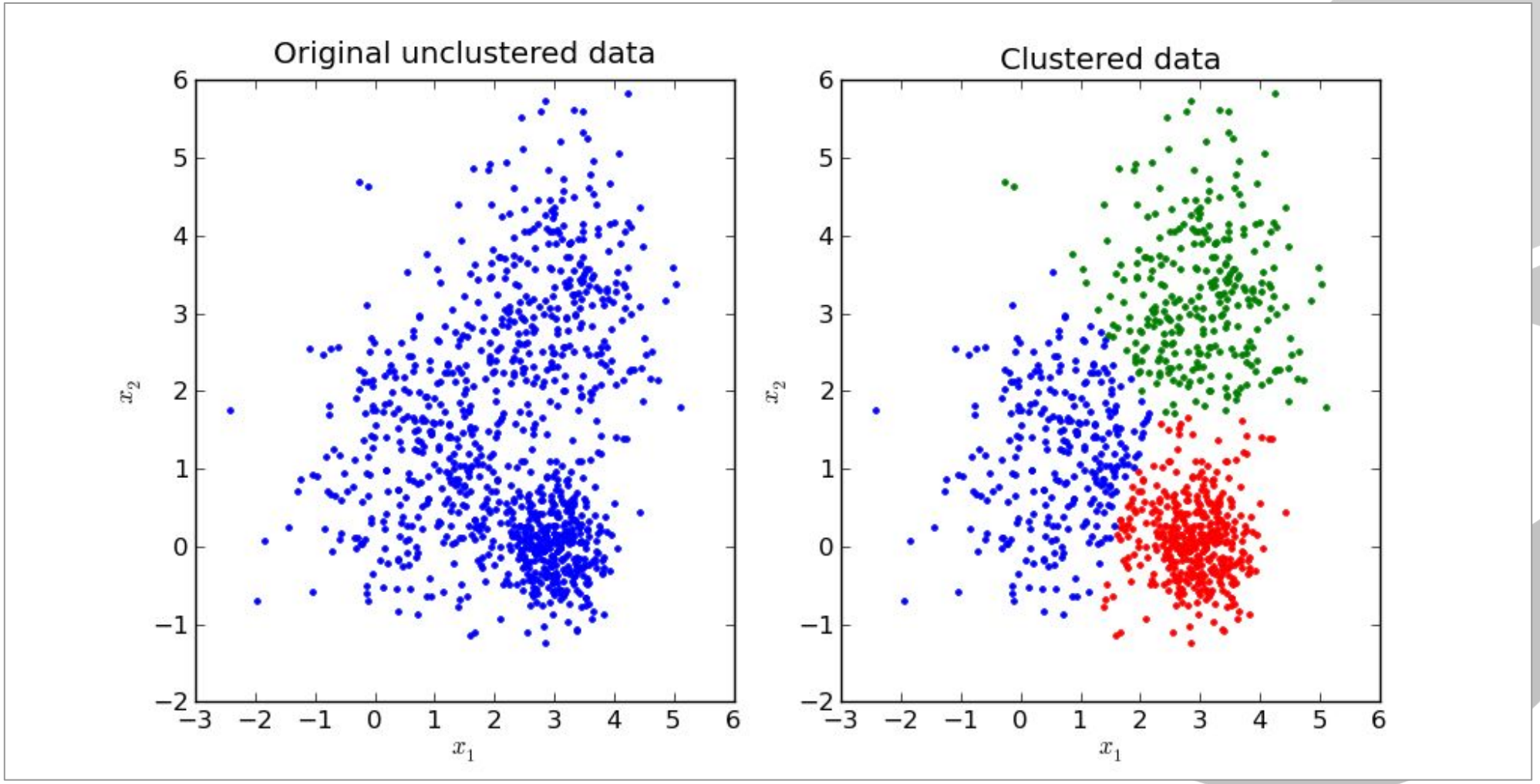
Tipos de Aprendizagem

Aprendizagem Supervisionada	Aprendizagem Não-Supervisionada	Aprendizagem por Reforço
Regressão	Clusterização	Q-Learning
Classificação	Filtragem Coletiva	SARSA
Redes Neurais Artificiais	Detecção de Anomalias	Deep Reinforcement Learning

Supervisionada vs Não-Supervisionada



Supervisionada vs **Não-Supervisionada**



Fonte: http://pypr.sourceforge.net/

Regressão Linear

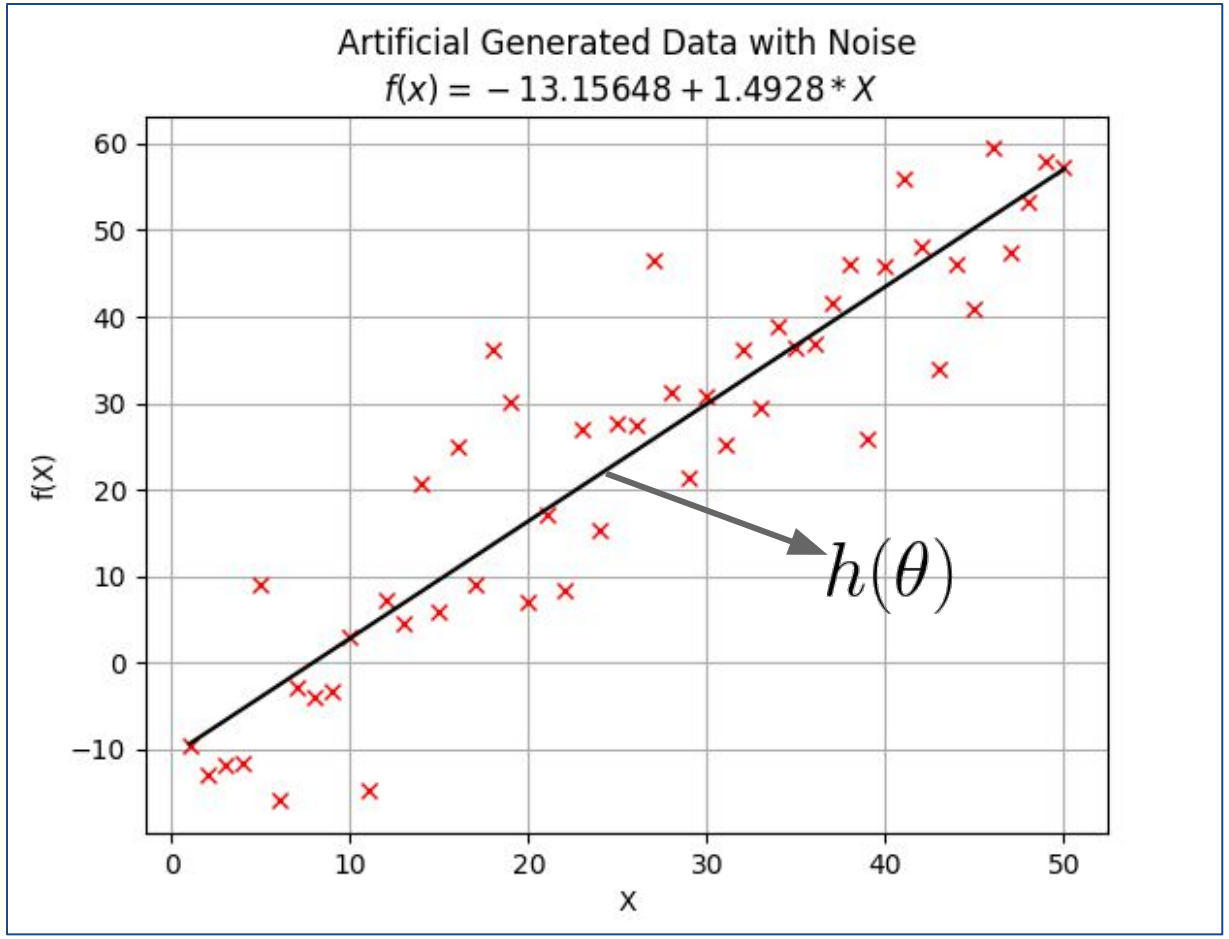
A mais simples e simpática técnica de Aprendizagem de Máquina

Regressão

- Consiste em técnicas de aproximar uma *variável dependente (X)* interpretando de *uma ou mais variáveis independentes (Y)*.

Modelo Resultante	Aplicação
parâmetros de uma função do tipo: $X \rightarrow Y$	estimar uma **função contínua** que não podem ser obtidas analiticamente.

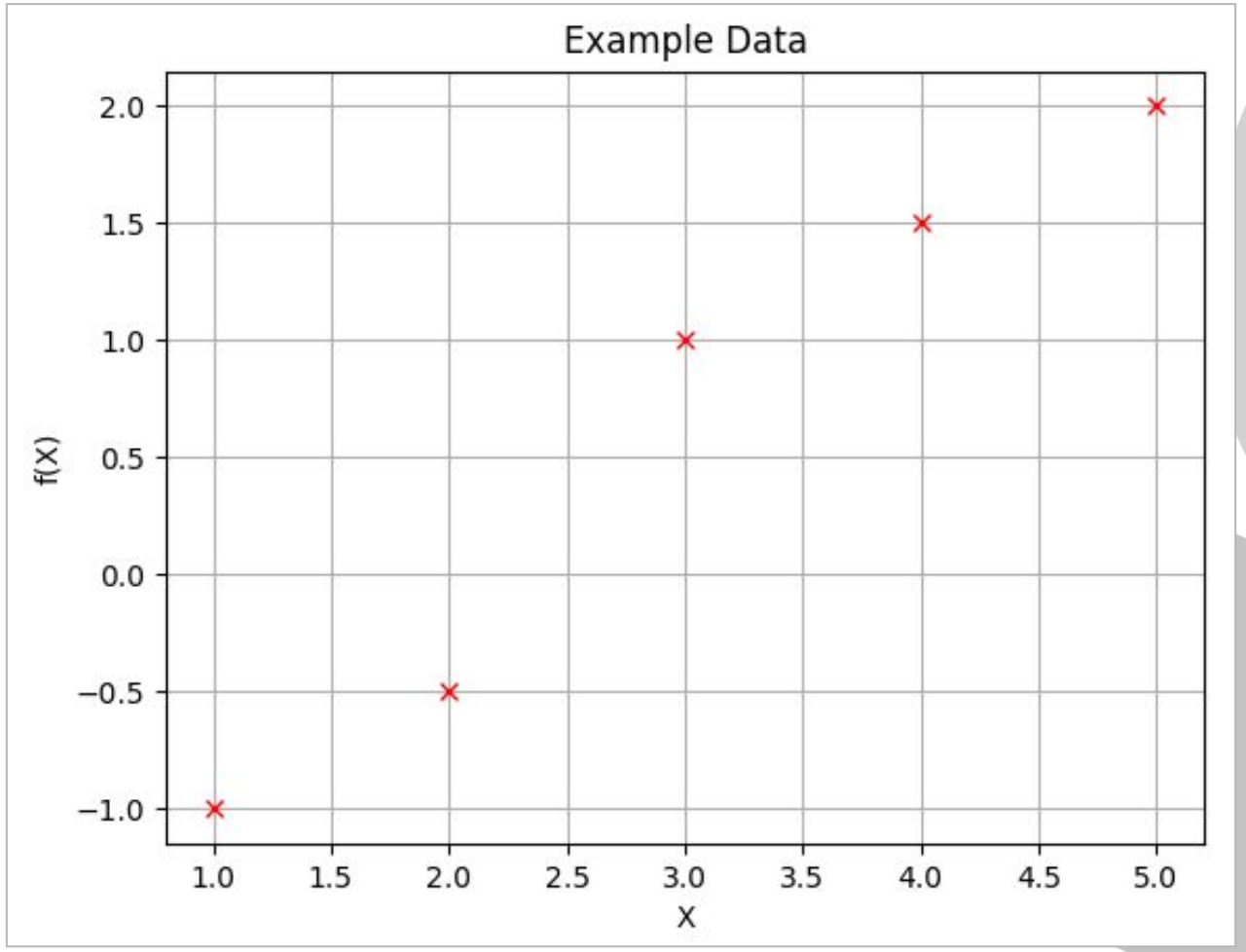
Regressão Linear



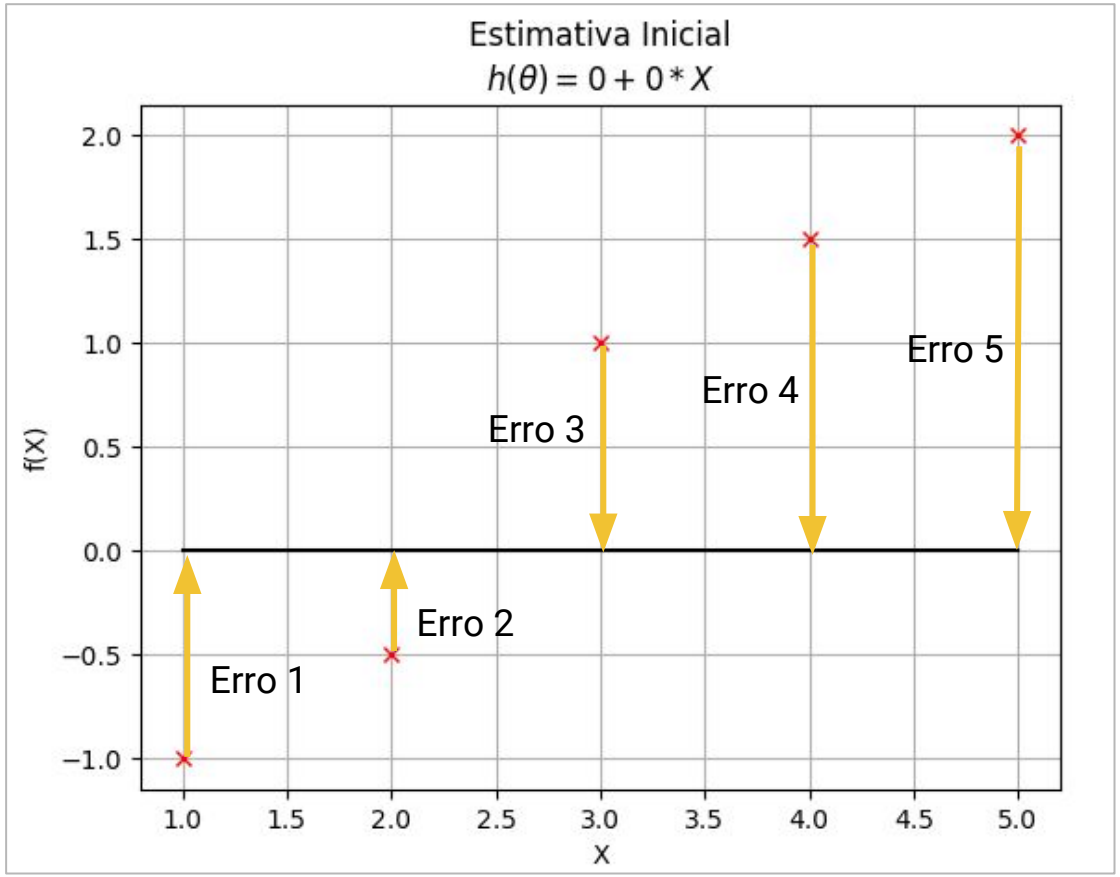
- Caso específico de Regressão, que usa um modelo **linear nos parâmetros**.
- Informalmente, o objetivo é "alinhar uma reta ao conjunto de dados".
- Modelo mais simples (é uma reta):

$$h(\theta) = \theta_0 + \theta_1 X_1$$

Cálculo do Custo



Cálculo do Custo (Estimativa Inicial)



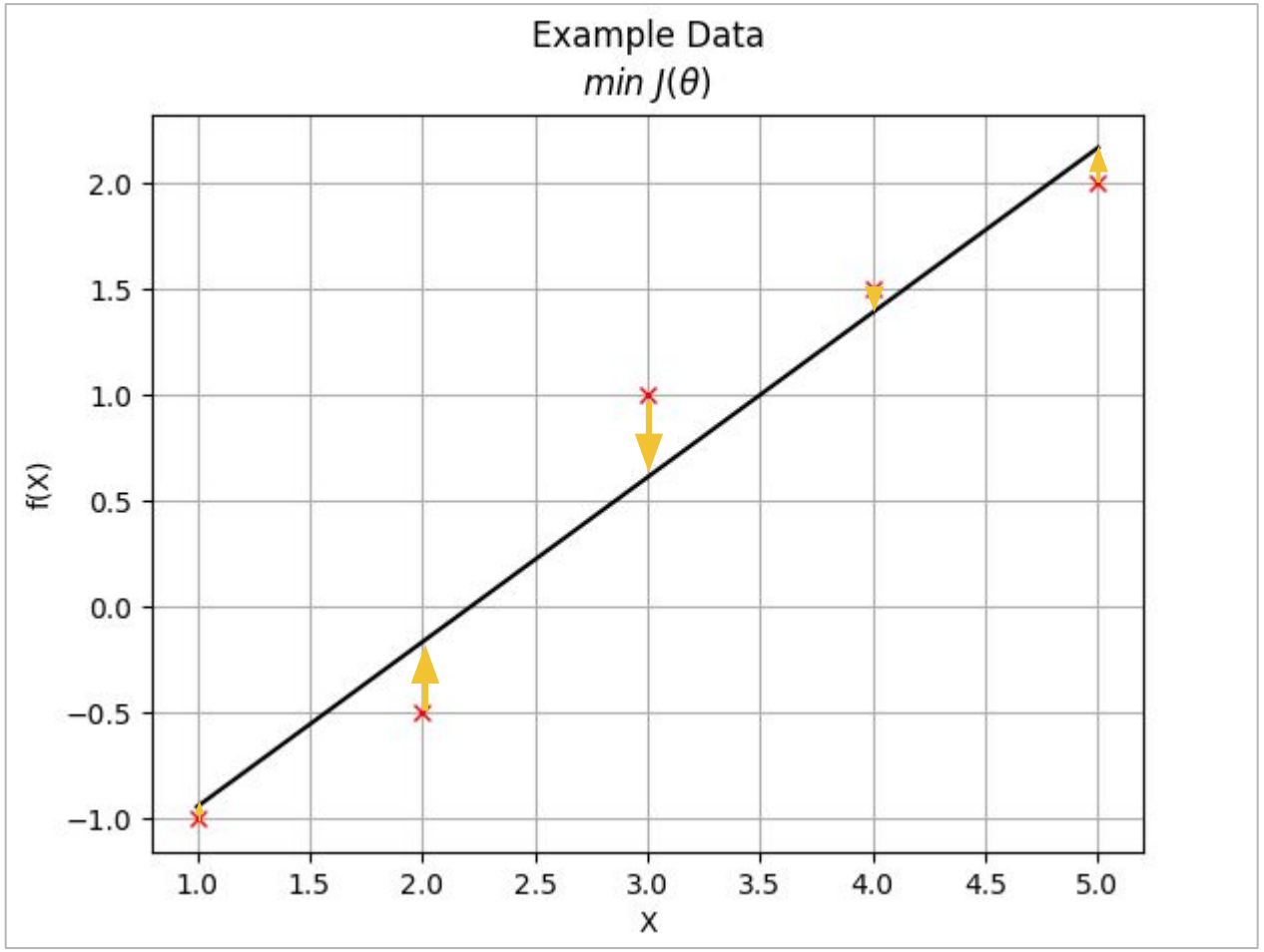
Residual Square Sum

$$J(\theta) = \sum_{i=1}^{5} (h_\theta^{(i)} - y^{(i)})^2$$

Objetivo de Otimização

$$\min_{|\theta|} J(\theta)$$

Cálculo do Custo



Esse modelo é o que minimiza os custos.

Mas, como descobrimos esses parâmetros?

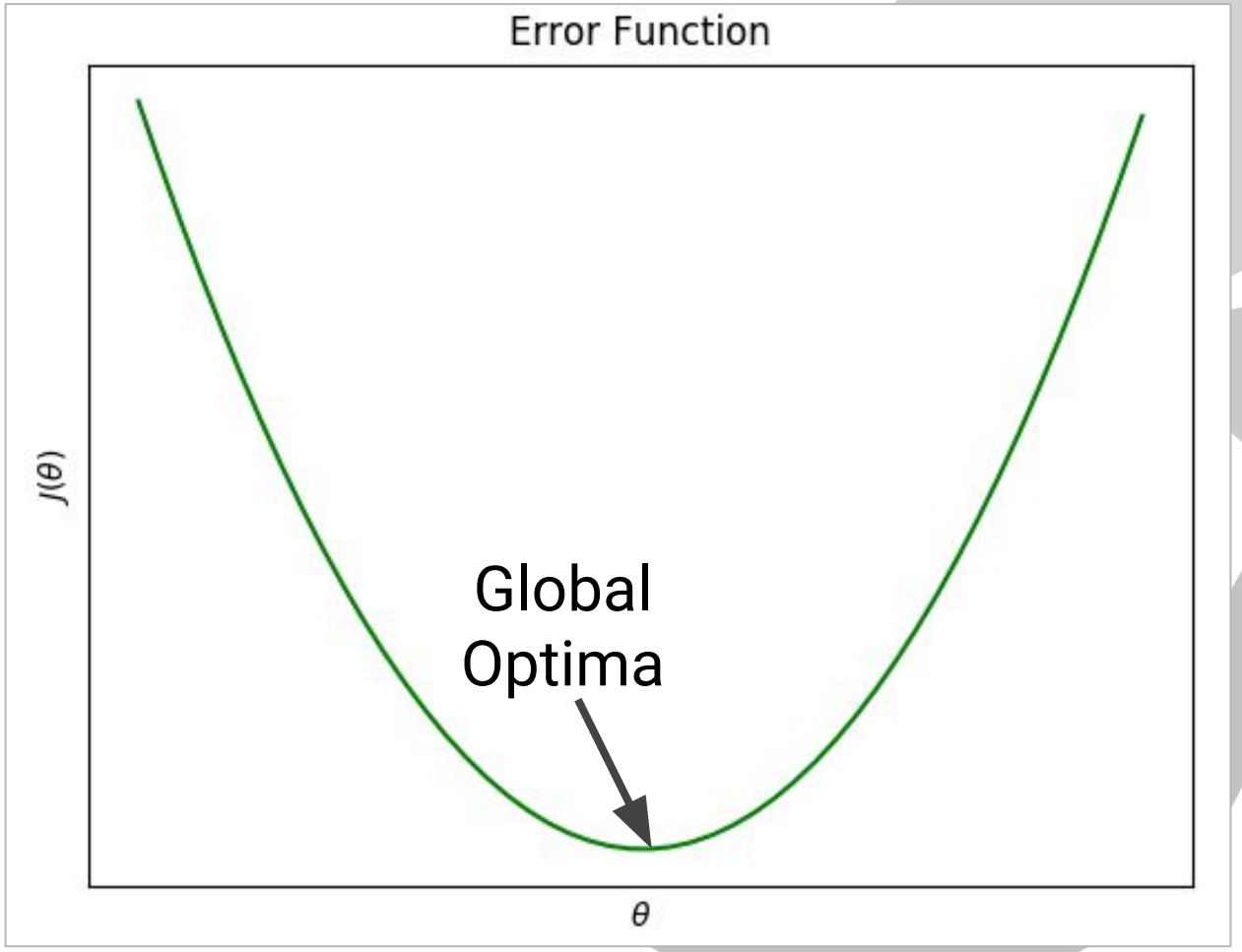
Gradiente Descendente

O famoso algoritmo de Otimização Convexa

Função do Custo Total*

A Função de Erro da Regressão Linear é uma função quadrática **convexa**!

*tridimensional



Global Optima

Q: Como calculamos valor que minimiza uma função? 🤔

A: Derivadas!

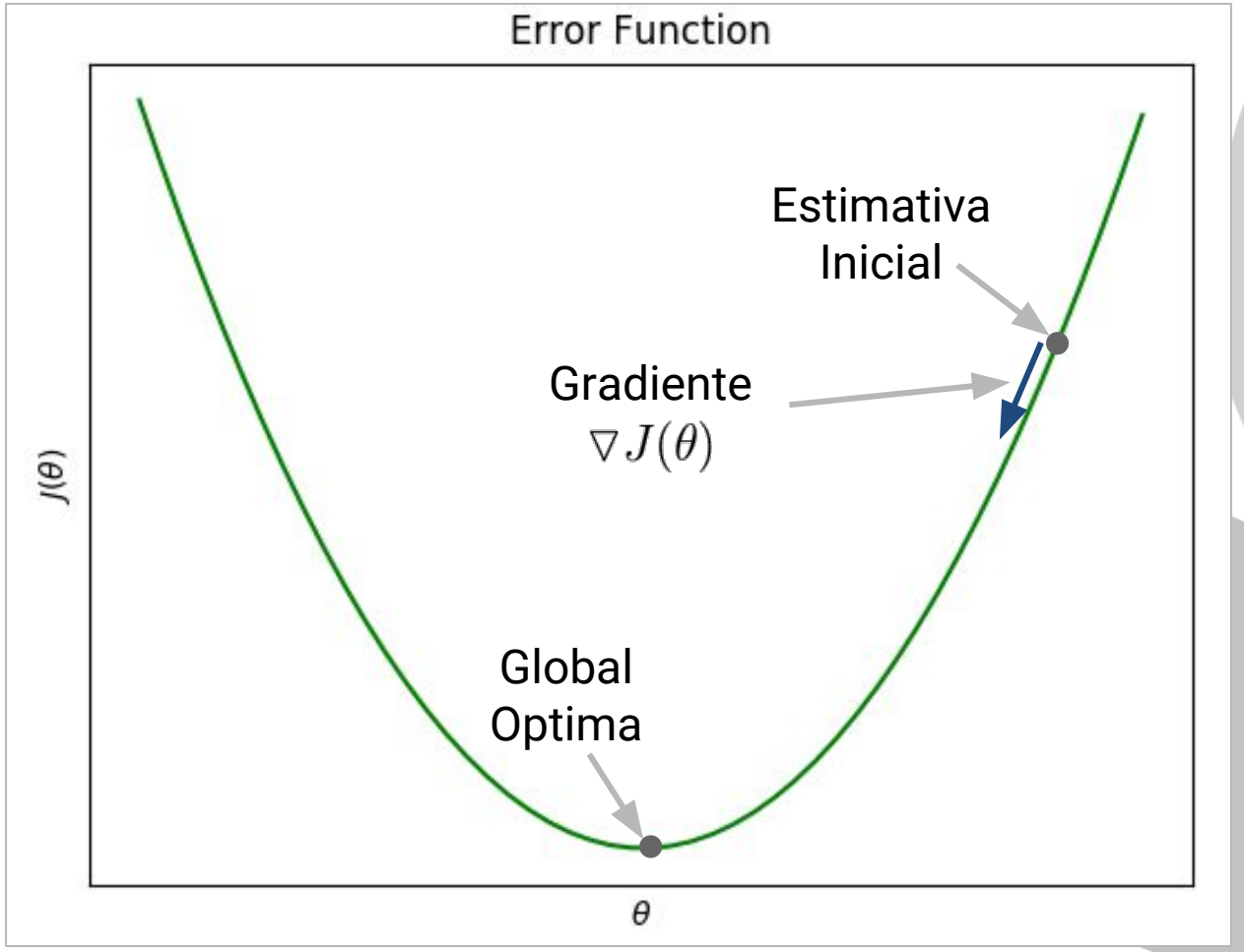
$$\frac{\delta(J(\theta))}{\delta\theta_0} = \frac{\delta}{\delta\theta_0}(\sum((\theta_0 + \theta_1 x) - y)^2 = \frac{1}{m}\sum(h(\theta) - y)$$

$$\frac{\delta(J(\theta))}{\delta\theta_1} = \frac{\delta}{\delta\theta_1}(\sum((\theta_0 + \theta_1 x) - y)^2 = \frac{1}{m}\sum(h(\theta) - y)x_1$$

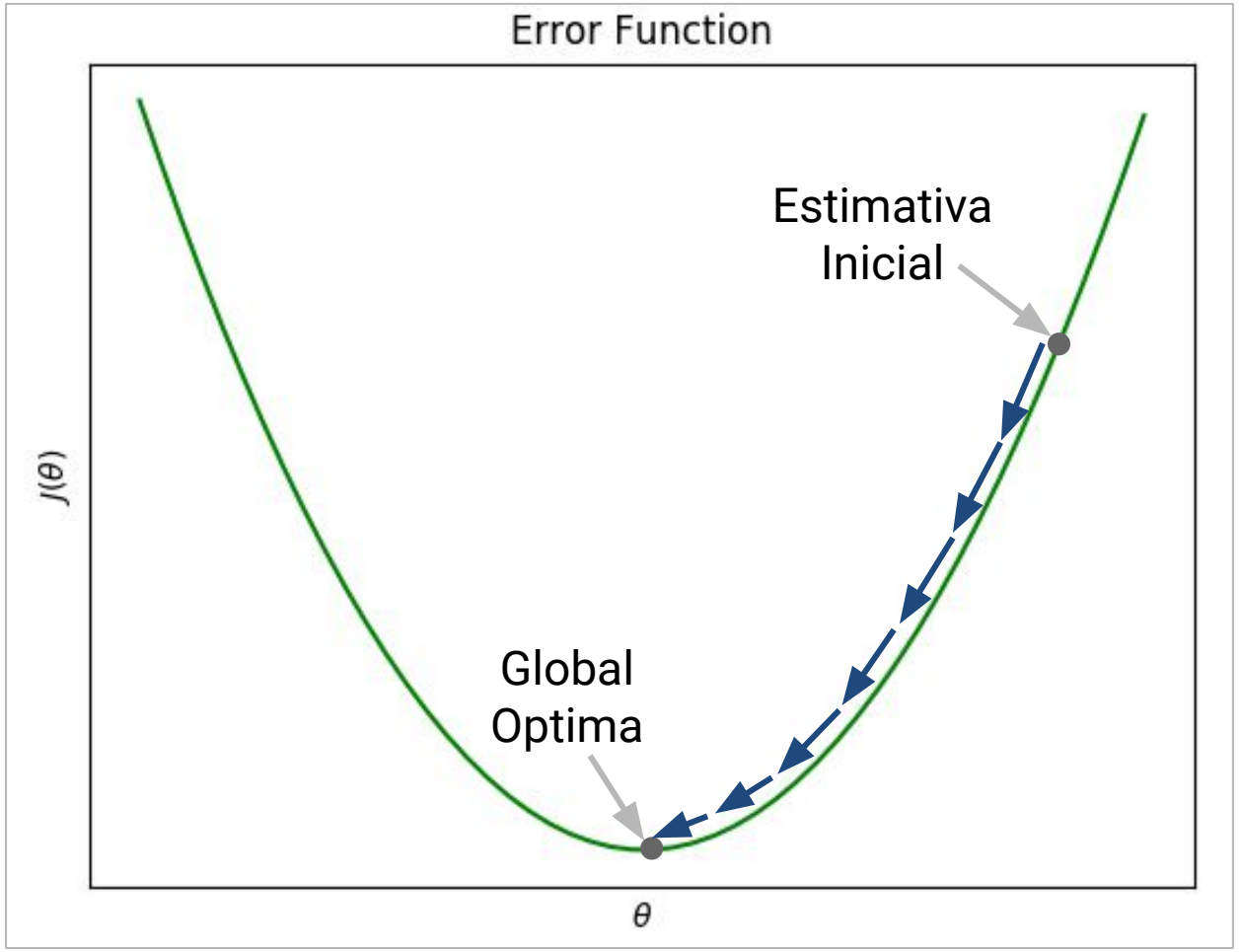
Matemágica

Matemágica

Vetores Gradientes



Gradiente Descendente



$$\nabla J(\theta) = (\frac{\delta J(\theta)}{\delta \theta_0}, \frac{\delta J(\theta)}{\delta \theta_1})$$

Algoritmo:

enquanto nao converge:

$$\theta_0 := \theta_0 - \frac{\alpha}{m} \sum (h(\theta) - y)$$

$$\theta_1 := \theta_1 - \frac{\alpha}{m} \sum (h(\theta) - y) x_1$$

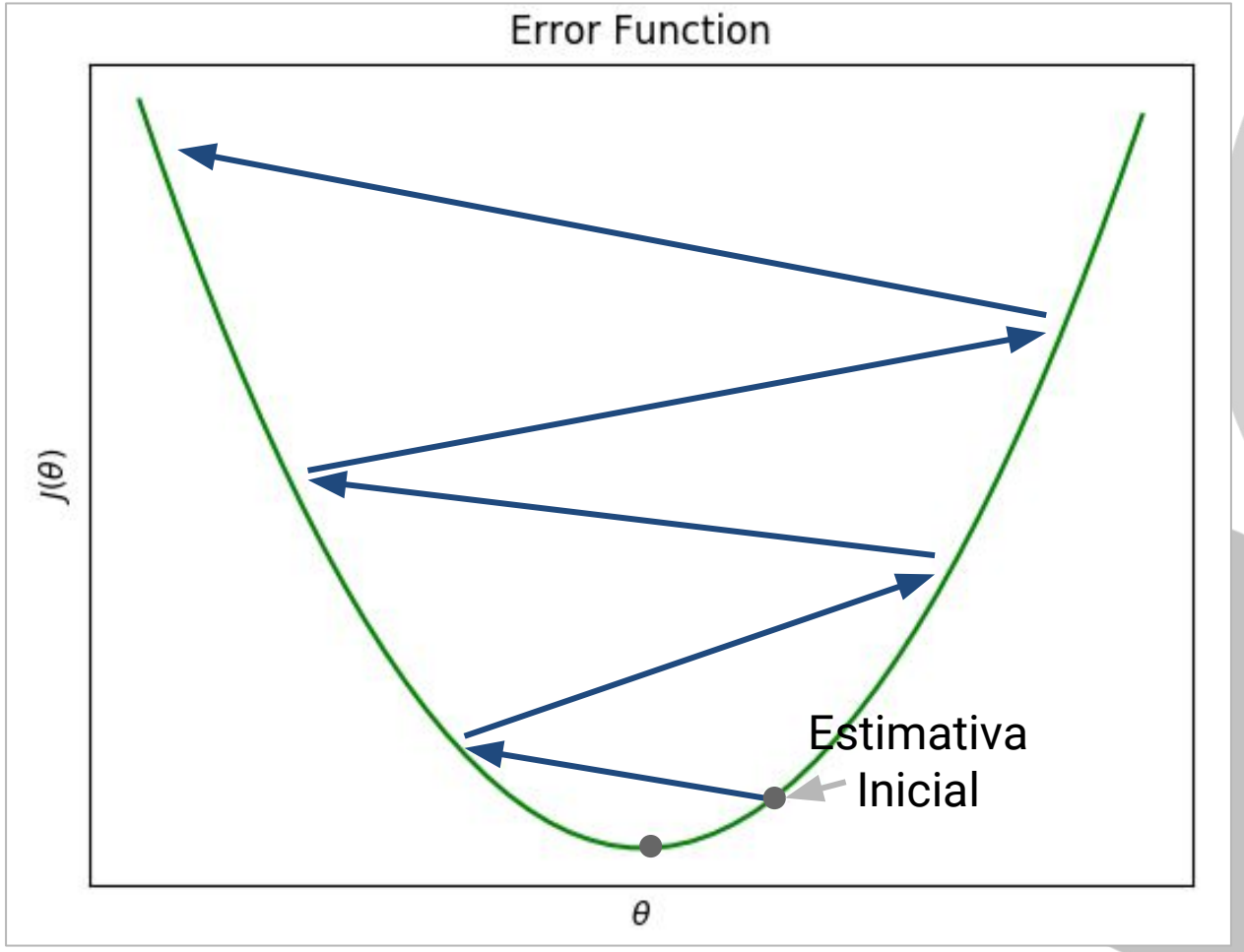
Detalhes Algorítmicos

- Cada **passo** do Gradiente Descendente é chamado de **época**.
- Criamos um atributo $x_0 = 1$ (*bias*) para que possamos resumir o Gradiente Descendente em:

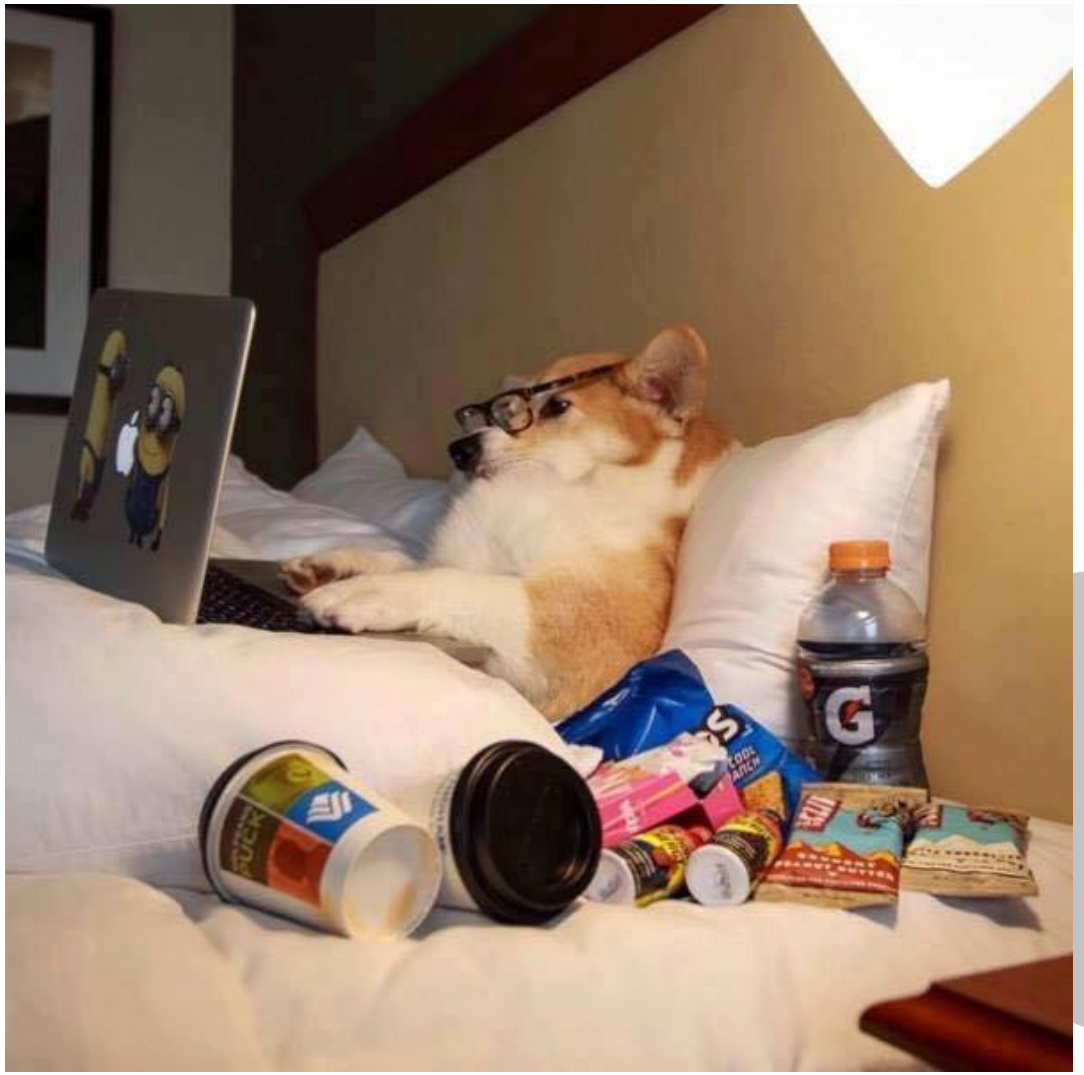
$$\text{enquanto nao converge:}$$
$$\theta_i := \theta_i - \frac{\alpha}{m} \sum (h(\theta) - y) x_i$$

- O "*Learning Rate*" (alfa) é um **hiperparâmetro**, e deve ser atribuído pelo próprio programador. Mas, tenha atenção...

Learning Rate Alto



Hands-On!!!



Obrigado!

minhotmog@gmail.com

www.petcomp.ufc.br