

Análise Estatística de dados

Inteligência Artificial



AULA 05 – OTIMIZAÇÃO

Arturo Forner-Cordero
Larissa Driemeier

PROGRAMA DO CURSO

Aula	Data	Conteúdo da Aula
01	27/02	Aula Inaugural
02	05/03	Introdução ao Curso. Noções de Álgebra Linear , Geometria Analítica Parte I
03	12/03	Noções de Álgebra Linear , Geometria Analítica Parte II
04	19/03	Decomposição de valor singular (SVD)
05	26/03	Otimização: derivadas, derivadas parciais (operadores gradiente, Jacobiano, Hessiano e Laplaciano), algoritmos de gradiente
06	02/04	Variáveis independentes e não independentes. Estatística Descritiva e Indutiva. Definições de medidas de dispersão e tendência central.
07	09/04	Probabilidade e Teorema de Bayes
08	16/04	Modelos de probabilidade discretos
09	23/04	Modelos de probabilidade contínuos
10	30/04	Modelo de Markov. Modelo de Markov oculto.

ONDE ESTAMOS?

Como me sinto?



Que vimos? Ferramentas

▪ Básicas:

- Álgebra linear,
- Vetores e matrizes

▪ Interpretação geométrica

▪ Avançadas:

- Fatorização de matrizes
- SVD: Singular Value Decomposition

CAMINHO: Que vamos a ver hoje?

Normas:

- Medida de distâncias

Derivada:

- Como varia uma função

Gradiente:

- Derivadas multidimensionais
- Jacobiano
- Hessiana

Otimização:

- A melhor solução

Gradiente descendente:

- A procura do ótimo



NORMAS

L^1, L^2, L^∞

NORMA

Considere dois classificador tentando prever múltiplos valores (por exemplo, preço de um bem de mercado).

O primeiro prevê valores 1, 2, 3, 4, 5

O segundo prevê valores 2, 3, 3, 3, 4

Os valores reais foram 4, 3, 2, 3, 1.

Qual dos dois se saiu melhor? E por quanto?

NORMA

Os valores reais foram 4, 3, 2, 3, 1.

O primeiro prevê valores 1, 2, 3, 4, 5

- Erros -3, -1, 1, 1, 4

O segundo prevê valores 2, 3, 3, 3, 4

- Erros -2, 0, 1, 0, 3,

Como atribuir uma medida a estes vetores?

O QUE SÃO NORMAS DE UM VETOR?

Normas de um vetor são funções matemáticas que atribuem um valor não negativo a um vetor, representando sua magnitude ou tamanho.

Com as normas pode-se medir “distância” entre vetores, o que as torna essenciais em várias tarefas de aprendizado de máquina, como agrupamento, classificação e regressão.

As normas fornecem uma medida quantitativa da similaridade ou dissimilaridade entre vetores, permitindo-nos comparar e contrastar seus desempenhos.

PROPRIEDADES MATEMÁTICAS DE UMA NORMA

Positividade-definida:

$$\|x\| \geq 0 \text{ e } \|x\| = 0 \rightarrow x = 0$$

A norma é sempre não negativa. Mais do que isso, ela só é nula quando o vetor é nulo.

PROPRIEDADES MATEMÁTICAS DE UMA NORMA

Homogeneidade absoluta:

$$\|x + y\| \leq \|x\| + \|y\|$$

A norma da soma de dois vetores é menor ou igual à soma dos dois vetores.

PROPRIEDADES MATEMÁTICAS DE UMA NORMA

Homogeneidade absoluta:

$$\|ax\| = |a|\|x\|$$

A norma do produto de um vetor por um escalar é a norma do vetor multiplicada pelo valor absoluto do escalar.

NORMA

De maneira geral, medimos o tamanho dos vetores usando uma função chamada norma L^p .

$$\|x\|_p = \left(\sum_i |x_i|^p \right)^{1/p}$$

para $p \in \mathbb{R}, p \geq 1$.

NORMA L^1

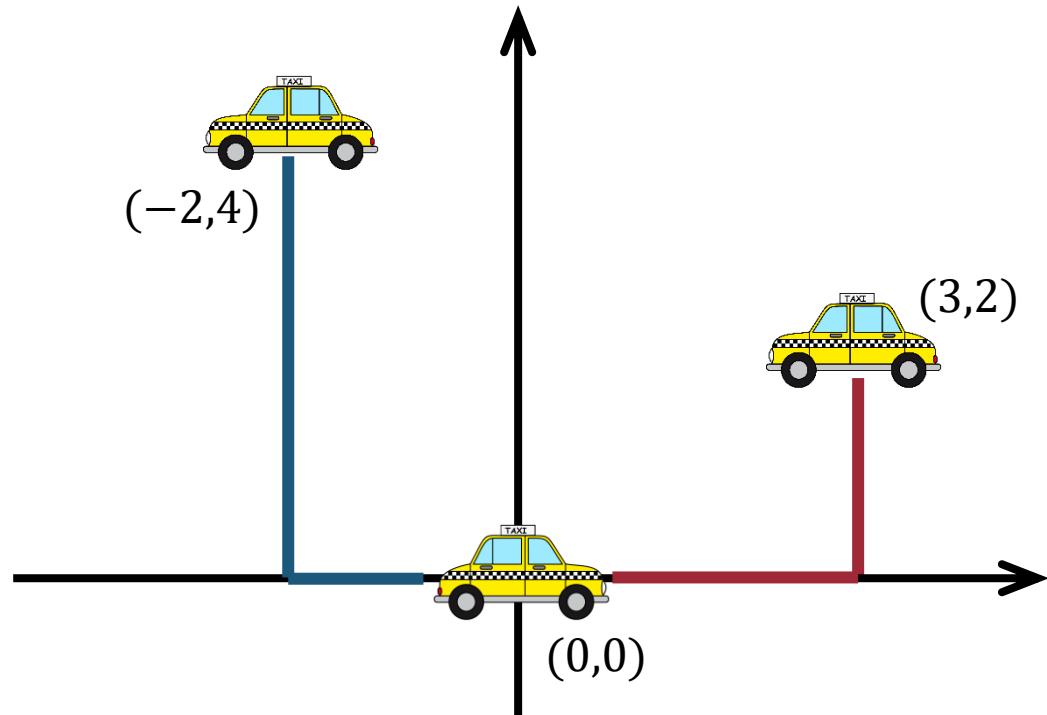
$$\|x\|_1 = \sum_i |x_i|$$

```
#Norma L1
v = np.array([-1, -2, 3, 4, 5])
np.linalg.norm(v, 1)
```

Norma L1 tem vários outros nomes... métrica do táxi, distância L1 ou distância de Manhattan.

Basicamente é a soma das magnitudes dos vetores em um espaço.

NORMA L^1 (MANHATTAN)



Veja que todos os componentes do vetor têm o mesmo peso.

A norma L1 representa a distância percorrida a partir da origem (0,0) até o destino (3,2) ou (-2,4) de maneira semelhante à forma como um táxi navega entre quarteirões da cidade para chegar ao seu destino.

$$\|x\|_1 = \sum_i |x_i| = 3 + 2 = 5$$

$$\|x\|_1 = \sum_i |x_i| = 2 + 4 = 6$$

EXEMPLO

```
X, Y = np.meshgrid(np.arange(-2, 2, .1), np.arange(-2, 2, .1))
print('vetor X\n', X, '\n')
print('vetor Y\n', Y, '\n')
```

EXEMPLO

```
X, Y = np.meshgrid(np.arange(-2, 2, .1), np.arange(-2, 2, .1))
print('vetor X\n', X, '\n')
print('vetor Y\n', Y, '\n')

plt.scatter(X, Y, marker='o');
```

```
Z=[X, Y]
print(np.shape(Z))
```

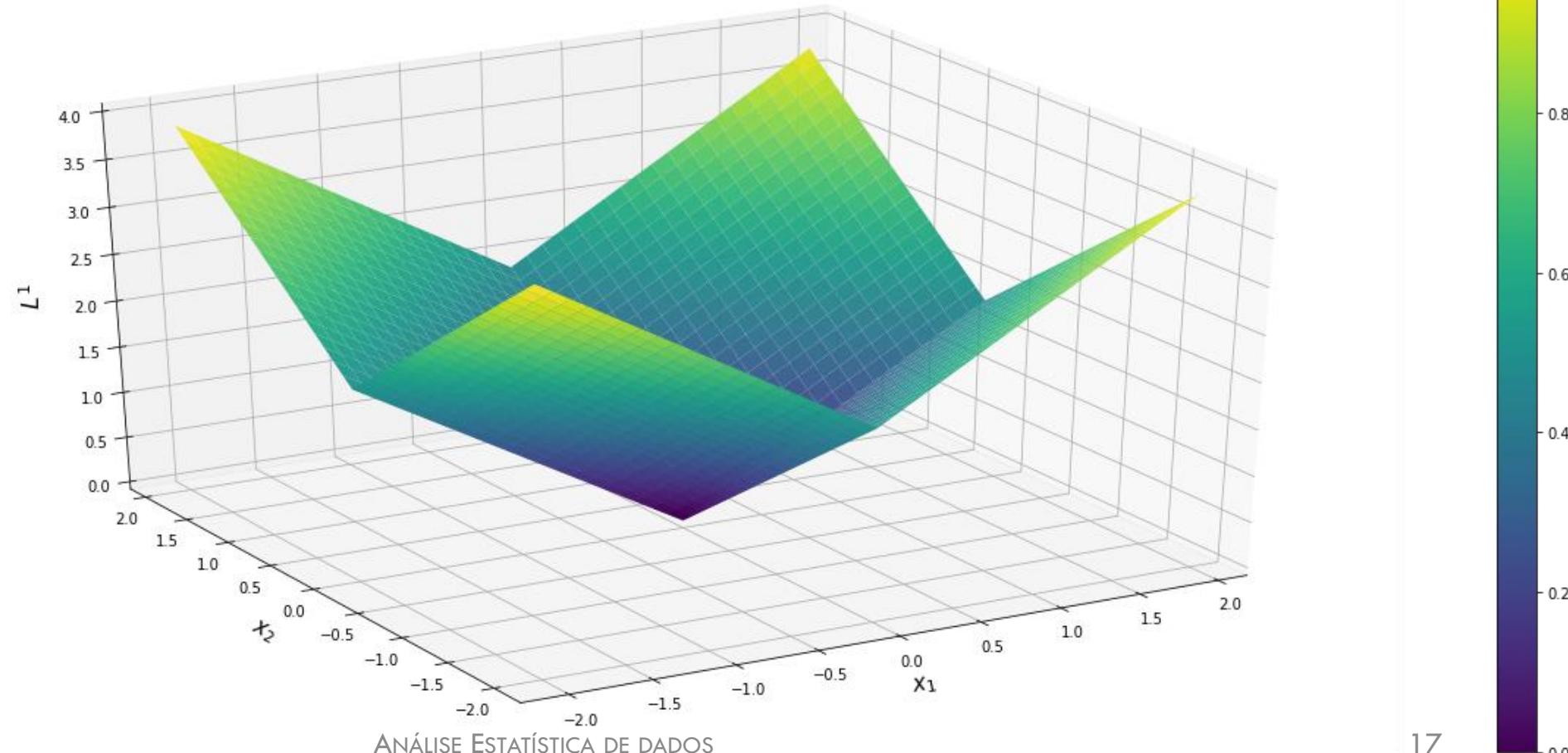
```
(2, 40, 40)
```

EXEMPLO

```
Z_L1 = np.linalg.norm(Z,1, axis=0) # np.abs(X)+np.abs(Y)
```

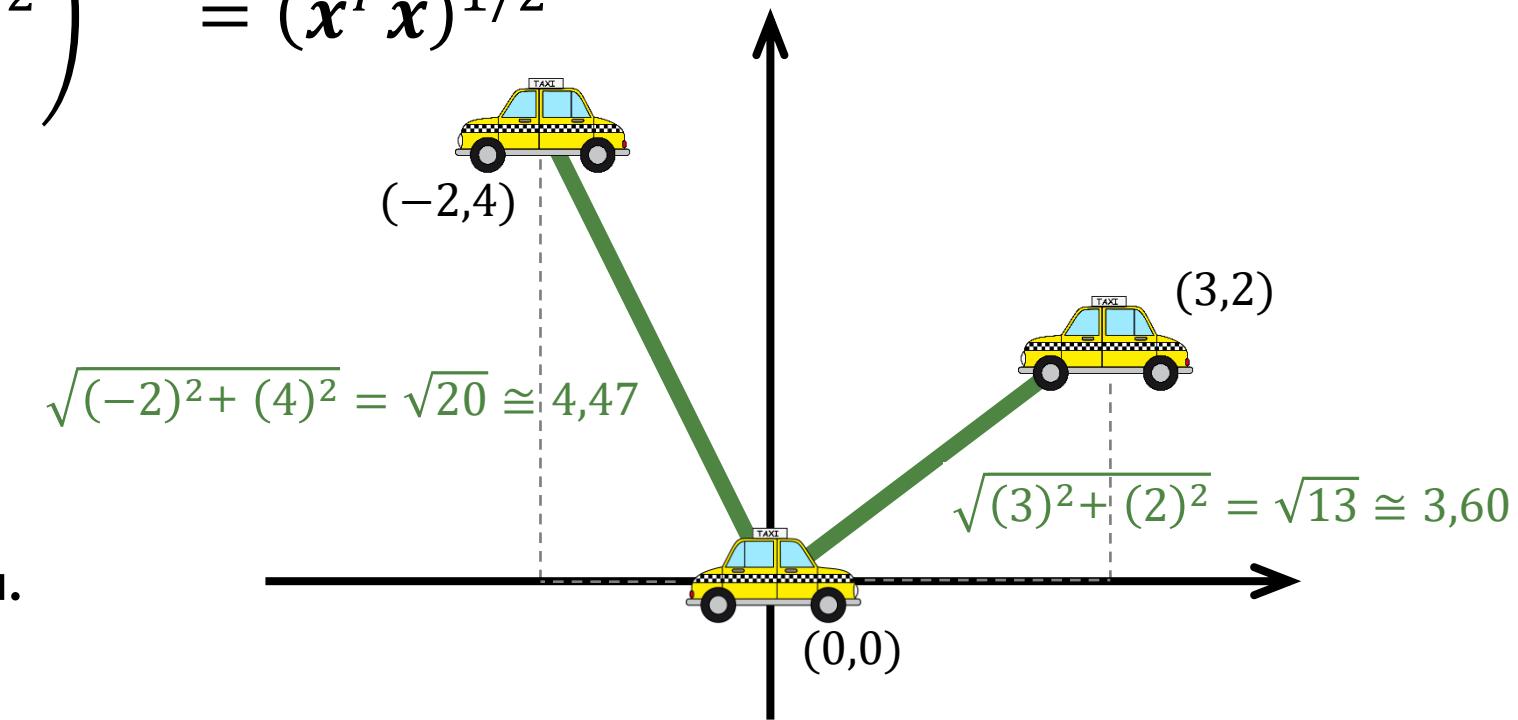
A norma L1 encontra aplicações em diversas tarefas de aprendizado de máquina. Uma aplicação proeminente é a regularização L1, também conhecida como regressão Lasso.

Mas, isso é assunto para o próximo ciclo...



NORMA EUCLIDIANA: L^2

$$\|x\|_2 = \|x\| = \left(\sum_i |x_i|^2 \right)^{1/2} = (x^T x)^{1/2}$$



A norma L2 é a rota mais direta.

NORMA EUCLIDIANA: L^2

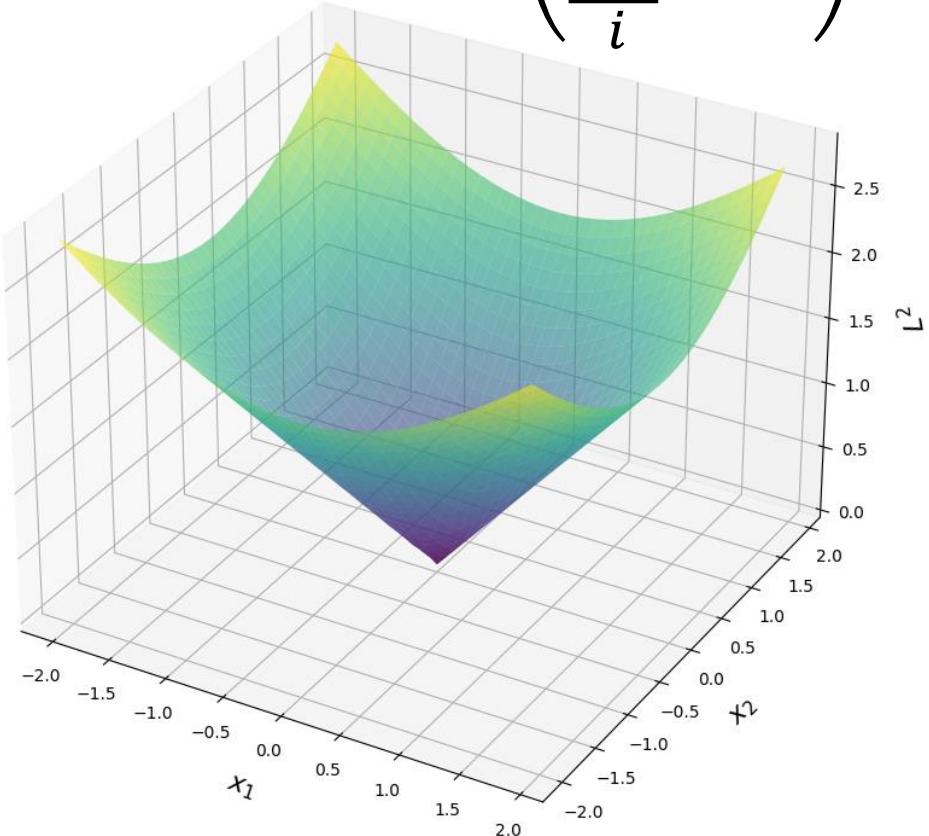
$$\|x\|_2 = \|x\| = \left(\sum_i |x_i|^2 \right)^{1/2} = (\mathbf{x}^T \mathbf{x})^{1/2} \quad \|x\|_2^2 = \|x\|^2 = \sum_i x_i^2 = \mathbf{x}^T \mathbf{x}$$

```

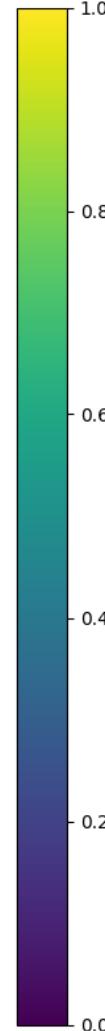
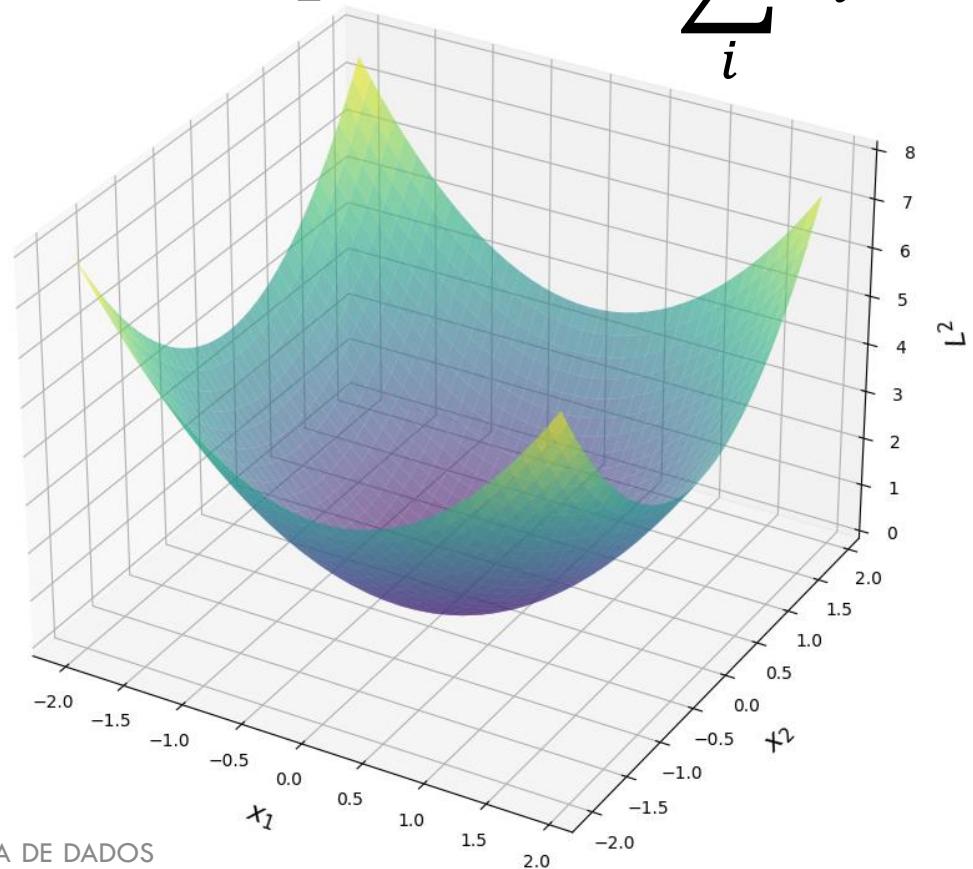
Z_L2 = np.linalg.norm(Z, 2, axis=0) # np.sqrt(X**2+Y**2)
Z_L2_2 = np.square(np.linalg.norm(Z, 2, axis=0)) # X**2+Y**2
  
```

NORMA EUCLIDIANA: L^2

$$\|x\|_2 = \|x\| = \left(\sum_i |x_i|^2 \right)^{1/2} = (x^T x)^{1/2}$$



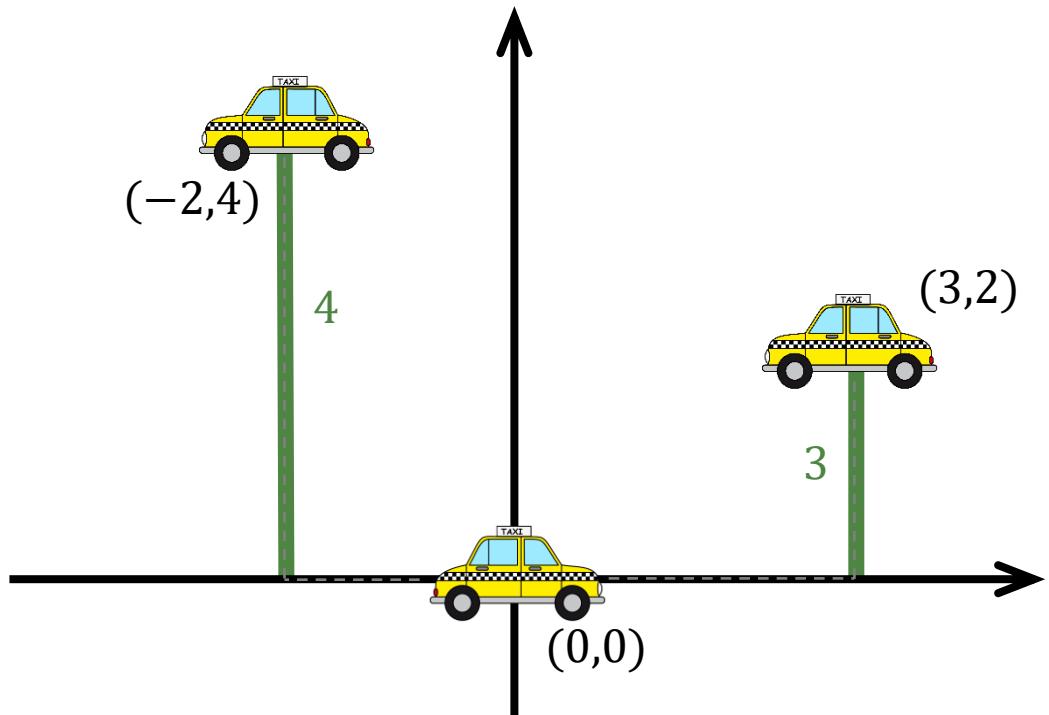
$$\|x\|_2^2 = \|x\|^2 = \sum_i x_i^2 = x^T x$$



NORMA L^∞

$$\|x\|_\infty = \max_i |x_i|$$

É o valor absoluto do maior componente do vetor. Portanto, também é chamada de **norma máxima**. Ou distância de Chebyshev.

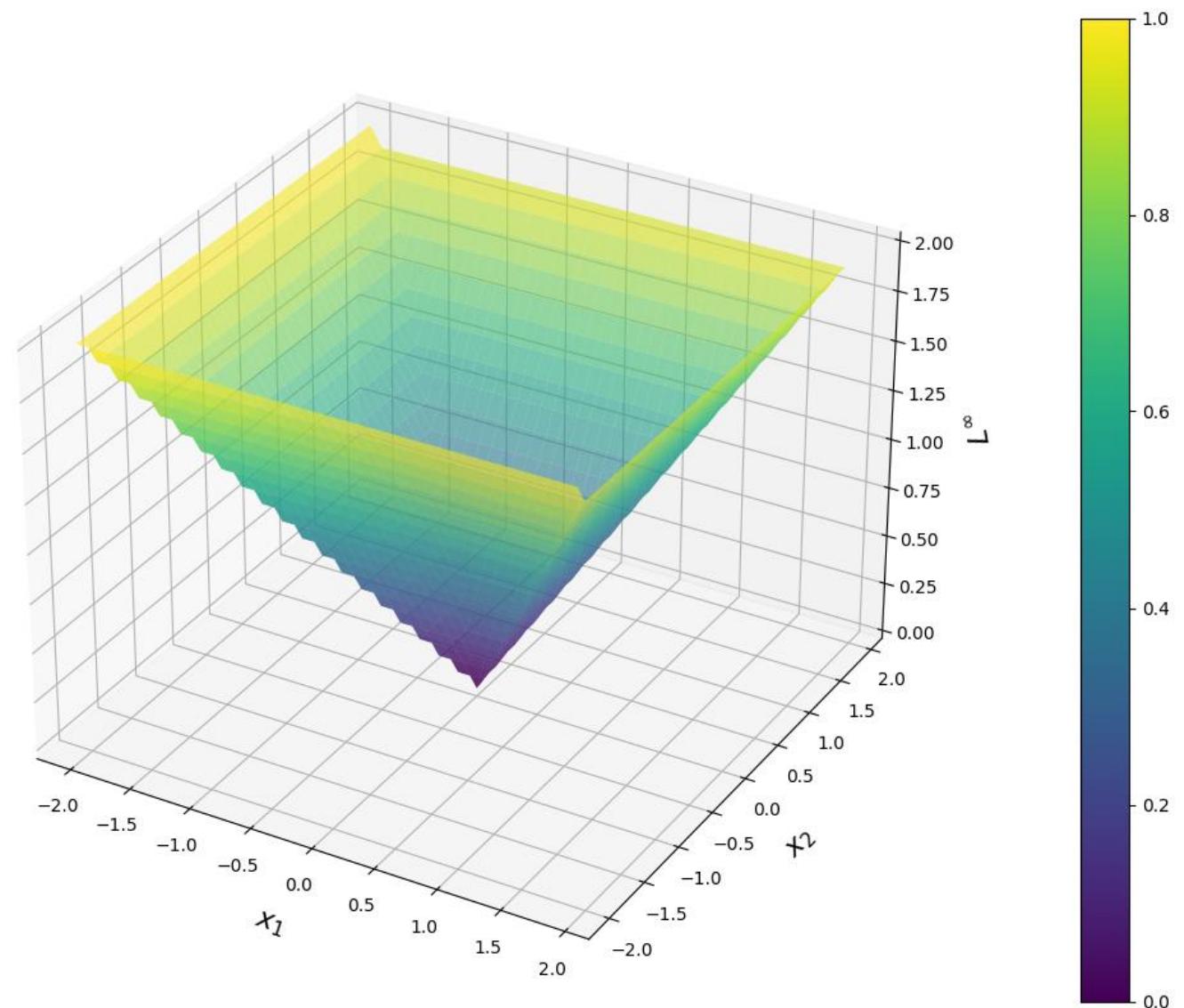


```
Z_inf = np.linalg.norm(Z,np.inf, axis=0) # np.amax([np.absolute(X),np.absolute(Y)], axis=0)
```

NORMA L^∞

$$\|x\|_\infty = \max_i |x_i|$$

É o valor absoluto do maior componente do vetor. Portanto, também é chamada de **norma máxima**. Ou distância de Chebyshev.

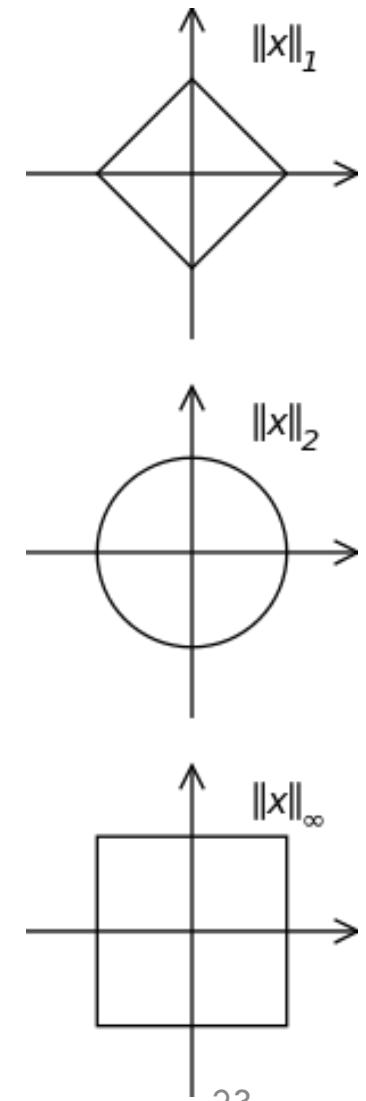


VIZUALIZANDO OS CÍRCULOS DA NORMA p

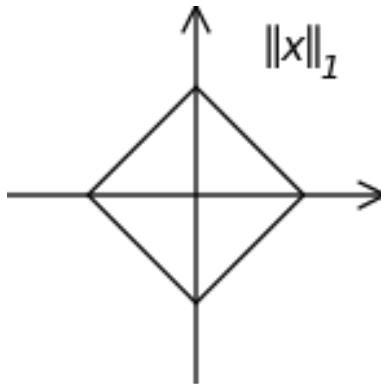
A norma L1 é formalmente definida como a soma do valor absoluto das coordenadas de um vetor. Então, por que o diamante?

A norma L2 é formalmente definida como o quadrado da diferença das coordenada de um vetor. Então, por que o círculo?

A norma L ∞ é formalmente definida como a dimensão absoluta máxima das coordenada de um vetor. Então, por que o quadrado?

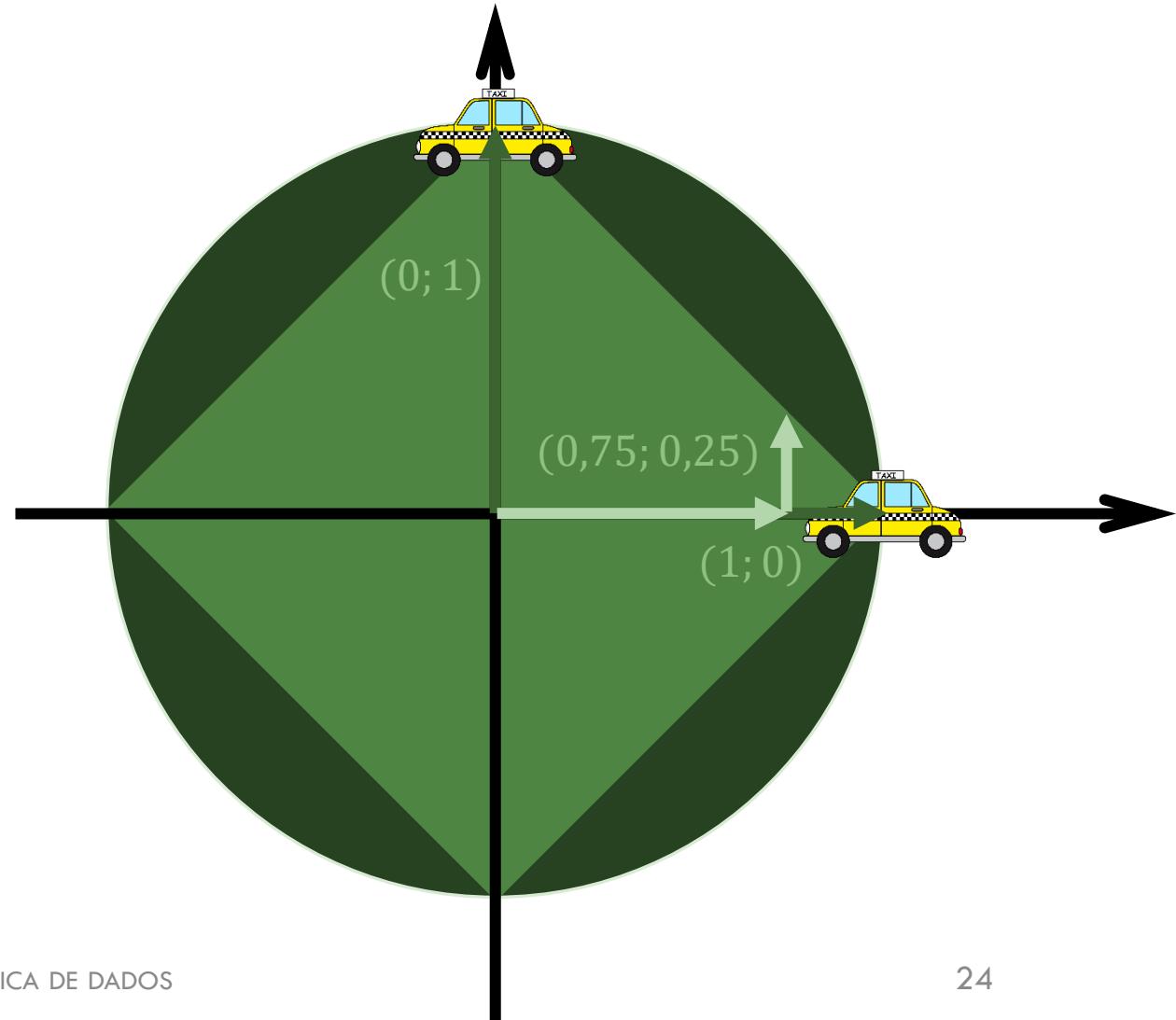


DIAMANTE L_1

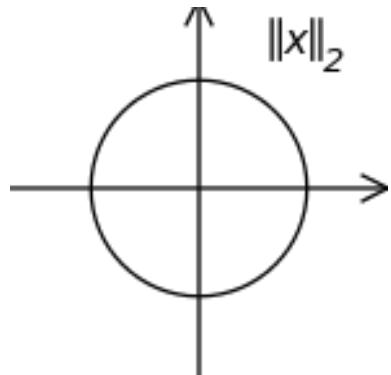


Um círculo é definido como um conjunto de pontos que estão a uma distância igual do centro.

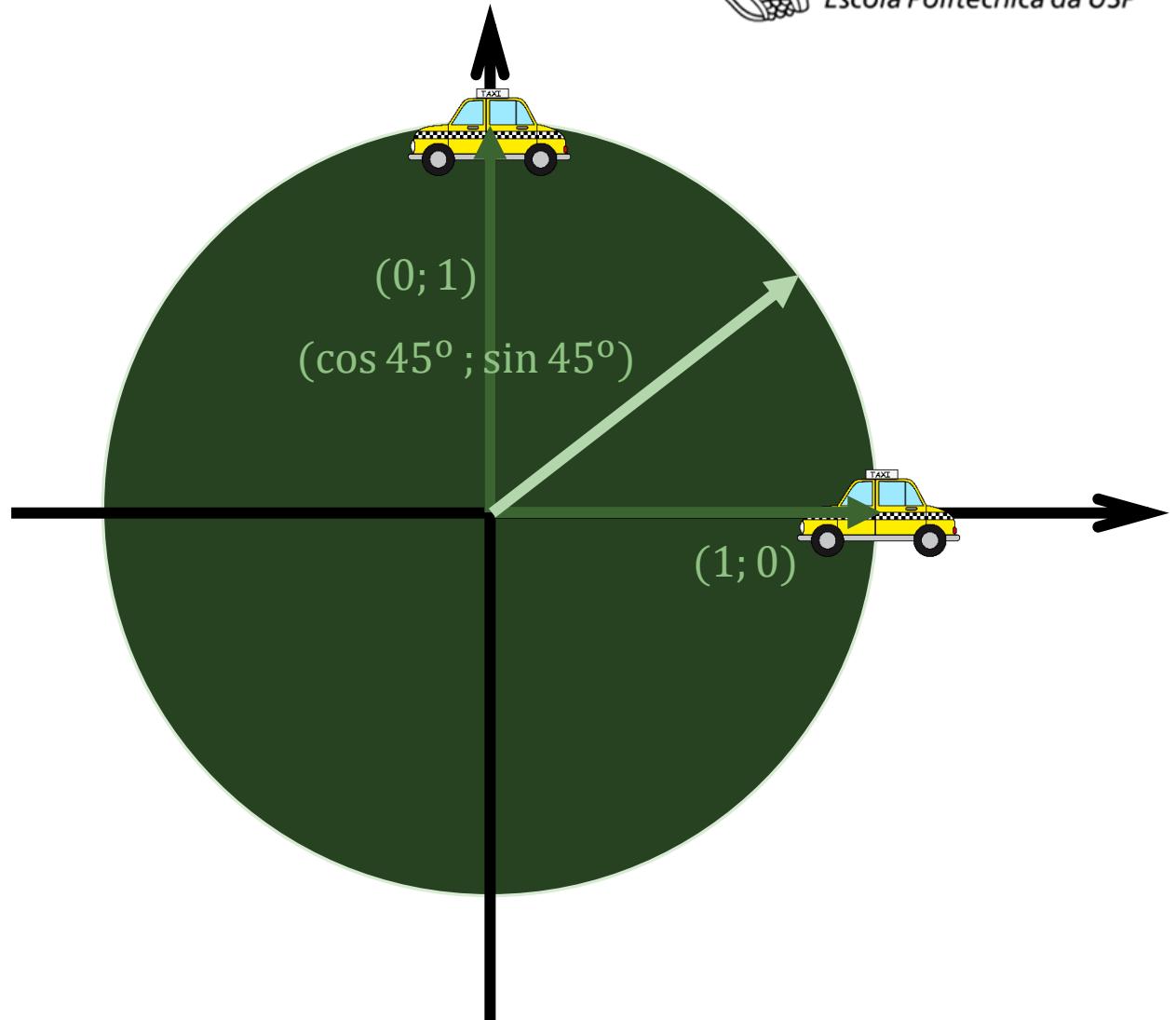
Se usarmos a distância de Manhattan para criar um círculo, obtemos este diamante. Imagine que a distância definida para este "círculo" seja 1. Um táxi poderia dirigir até $(1; 0)$ ou $(0,75; 0,25)$ ou $(0,01; 0,99)$ após percorrer 1 de distância no plano cartesiano. O diamante representa todos esses pontos possíveis.



CÍRCULO L_2

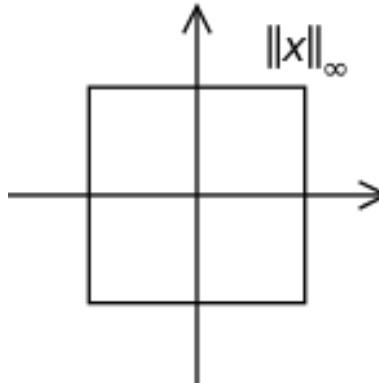


A forma circular faz mais sentido: a distância euclidiana nos permite seguir trajetórias em linha reta de ponto a ponto, permitindo-nos alcançar “mais profundamente” os cantos do diamante L_1 .

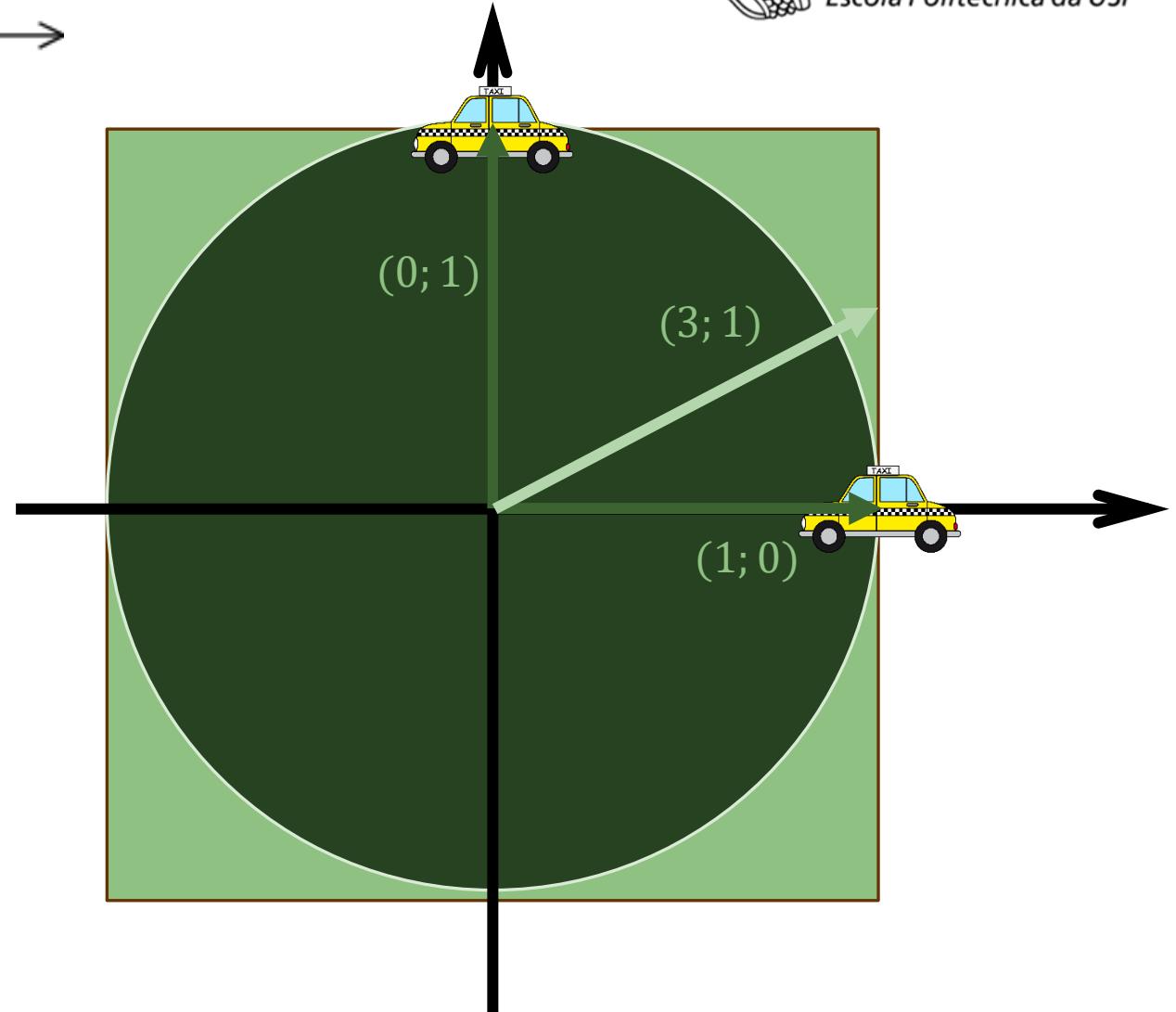


Um círculo desenhado com a definição de distância euclidiana é muito familiar.

QUADRADO L_∞



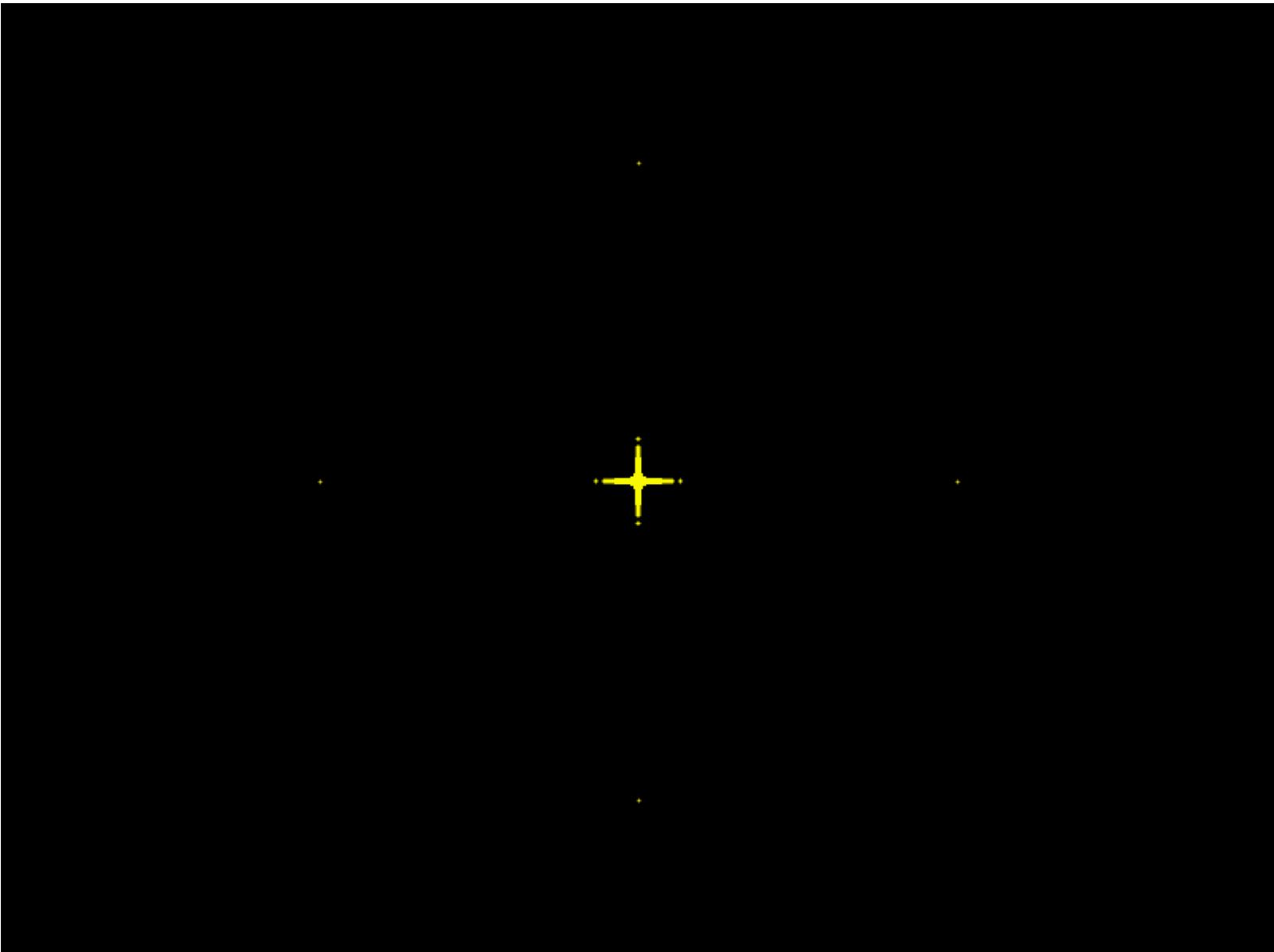
Se quisermos construir um círculo de distância 1, onde cada ponto tenha a mesma norma L_∞ ao centro, devemos escolher a coordenada absoluta mais alta: X ou Y . Portanto, todos os pontos a uma distância de 1 da origem terão $X = \pm 1$, $Y = \pm 1$, ou ambos, mas nunca mais do que isso. Queremos que a função de distância sempre escolha 1 como sua coordenada máxima absoluta de X ou Y . Assim, o quadrado — todos os pontos cuja coordenada máxima é 1.



EXEMPLO DO NOTEBOOK

```
C_L1 = np.array([[1,0],[0.75,0.25],[0.5,0.5],[0.25,0.75],[0,1]])
C_L2 = np.array([[1,0],[0,1], [np.cos(np.deg2rad(30)),np.sin(np.deg2rad(30))],
                 [np.cos(np.deg2rad(45)),np.sin(np.deg2rad(45))],
                 [np.cos(np.deg2rad(60)),np.sin(np.deg2rad(60))]])
C_Linf = np.array([[1,0],[1,0.5],[1,1],[0.5,1],[0,1]])
print(np.linalg.norm(C_L1,1, axis=1))
print(np.linalg.norm(C_L2,2, axis=1))
print(np.linalg.norm(C_Linf,np.inf, axis=1))
```

Animação
para
norma p
de $p = 0,1$
até $p = 2$



NORMA FROBENIUS DE UMA MATRIZ

$$\|A\|_F = \sqrt{\sum_{i,j} a_{ij}^2} = \sqrt{Tr(AA^T)}$$

$$A = \begin{bmatrix} 2 & 9 & 8 \\ 4 & 7 & 1 \\ 8 & 2 & 5 \end{bmatrix}$$

$$AA^T = \begin{bmatrix} 2 & 9 & 8 \\ 4 & 7 & 1 \\ 8 & 2 & 5 \end{bmatrix} \begin{bmatrix} 2 & 4 & 8 \\ 9 & 7 & 2 \\ 8 & 1 & 5 \end{bmatrix} = \begin{bmatrix} 2 \times 2 + 9 \times 9 + 8 \times 8 = 149 \\ 149 & 79 & 74 \\ 79 & 66 & 51 \\ 74 & 51 & 93 \end{bmatrix}$$

$2 \times 2 + 9 \times 9 + 8 \times 8 = 149$
 $Tr(AA^T) = 149 + 66 + 93 = 308$
 $\sqrt{Tr(AA^T)} = \sqrt{308} \cong 17,55$

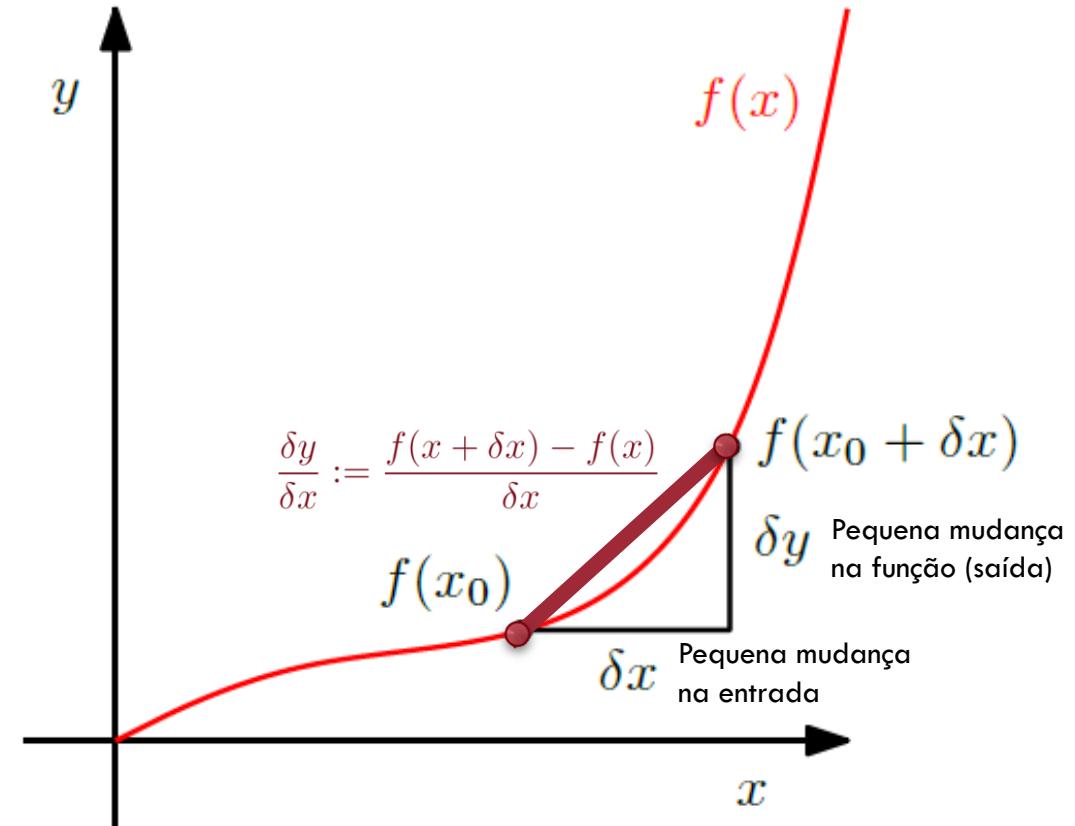


DERIVADA

O QUE É DERIVADA?

No cálculo, a derivada de uma função é basicamente o quanto a função muda se você alterar a entrada em uma quantia muito pequena,

$$\frac{\delta y}{\delta x} = \frac{f(x + \delta x) - f(x)}{\delta x}$$

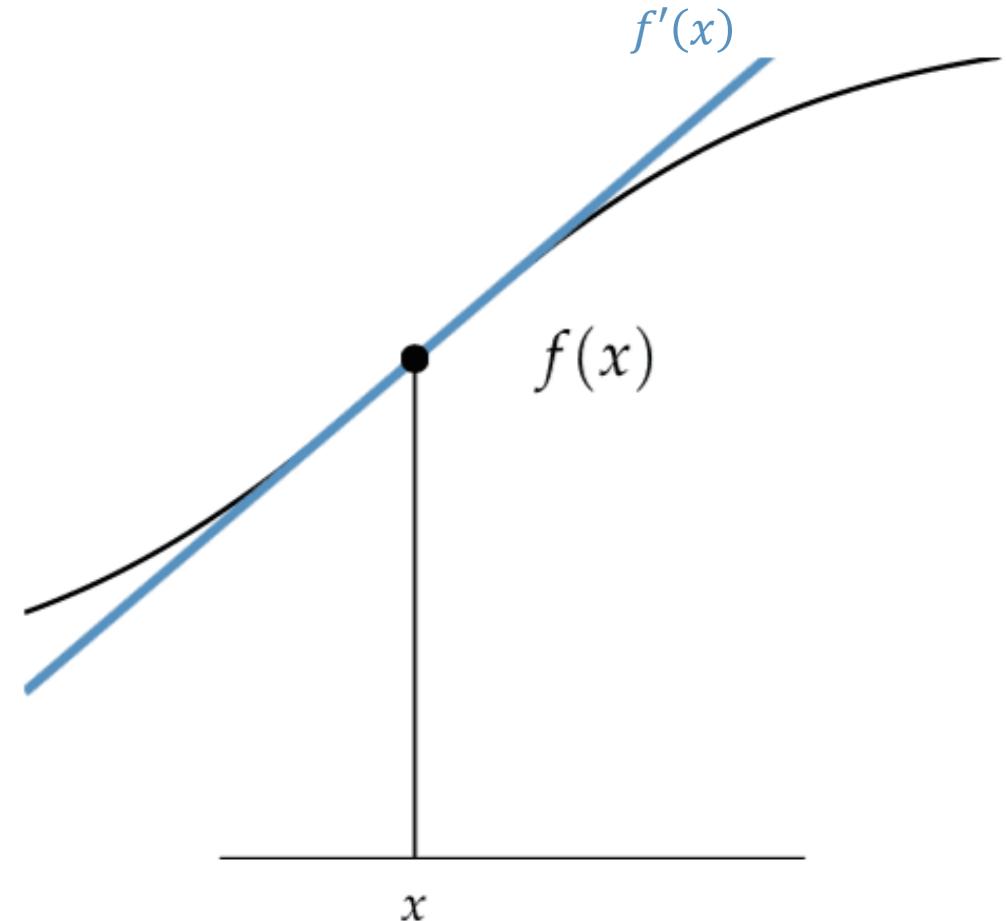


O QUE É DERIVADA?

No cálculo, a derivada de uma função é basicamente o quanto a função muda se você alterar a entrada em uma quantia muito pequena,

$$\frac{\delta y}{\delta x} = \frac{f(x + \delta x) - f(x)}{\delta x}$$

No limite para $\delta x \rightarrow 0$, obtemos a tangente de f em x . A tangente é então a derivada de f em x .



Uma função é diferenciável se pudermos calcular a derivada em todos os pontos de entrada para as variáveis da função. Nem todas as funções são diferenciáveis.

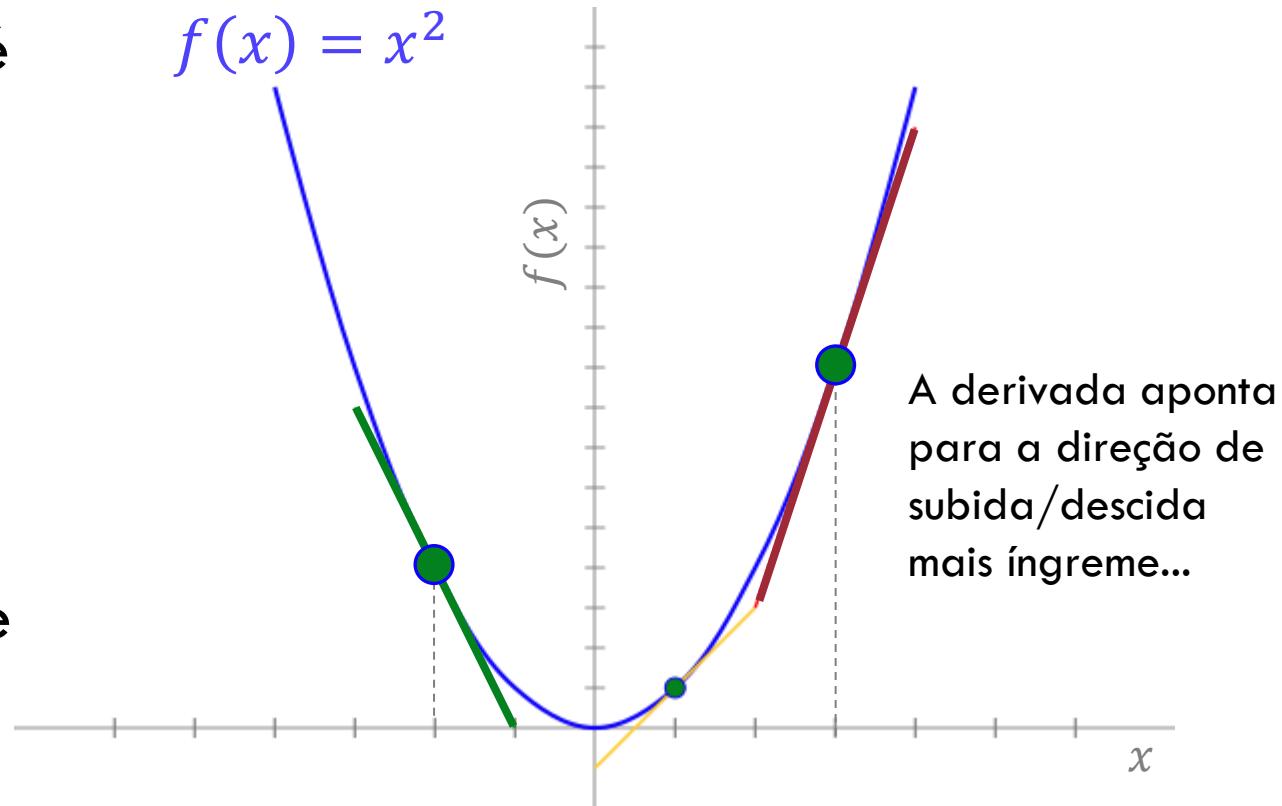
POR EXEMPLO

A derivada da função $f(x) = x^2$ é

$$f'(x) = \frac{df}{dx} = 2x$$

Então, a declividade de $f(x)$ em $x = 3$ é 6.

Veja que em $x = -2$ a declividade de $f(x)$ é negativa, e vale -4 .



VOCÊ NÃO PRECISA SABER DERIVAR...

```
import sympy
x = sympy.Symbol('x')
y = sympy.Symbol('y')
# Criando a equação
f = x * y + x ** 2 + sympy.sin(2 * y)
# Primeira derivada com respeito a x
df_dx = sympy.diff(f, x)
print("A derivada de f(x,y) com respeito a x é: " + str(df_dx))
# Segunda derivada comr espeito a y
d2f_dy2 = sympy.diff(f, y, 2)
print("A segunda derivada de f(x,y) com respeito a y é: " + str(d2f_dy2))
```

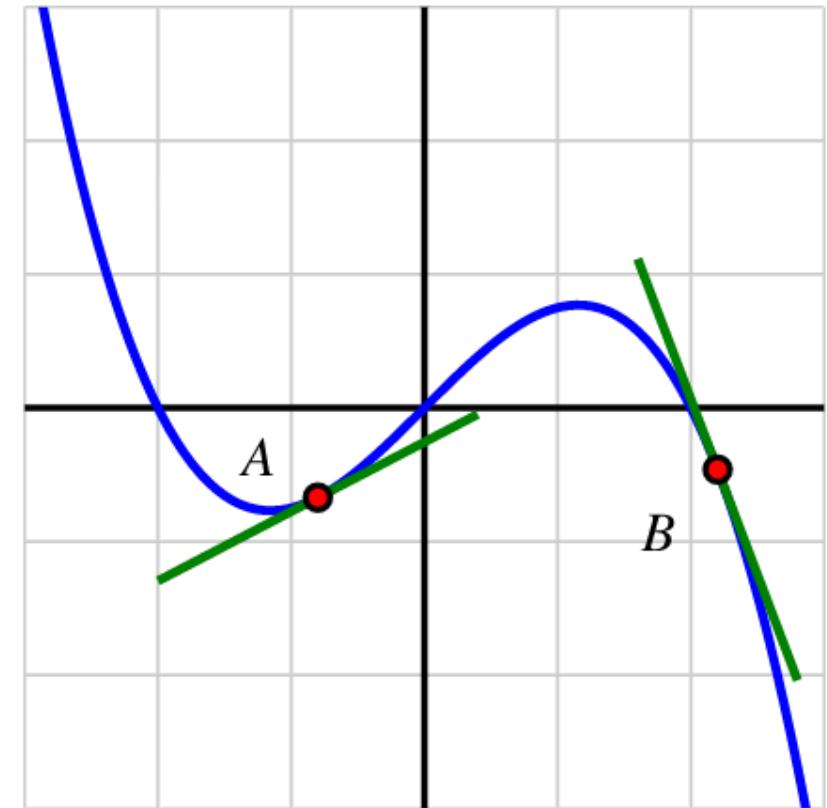
A derivada de f(x,y) com respeito a x é: $2*x + y$

A segunda derivada de f(x,y) com respeito a y é: $-4*\sin(2*y)$

Dada uma função diferenciável $f(x)$, sabemos que sua derivada $f'(x)$, é uma função cuja saída em $x = A$ nos diz a inclinação da reta tangente no ponto $(A, f(x = A))$.

A derivada de $f(x)$ nos diz não apenas se a função $f(x)$ está aumentando ou diminuindo em um intervalo, mas também **como** está aumentando ou diminuindo.

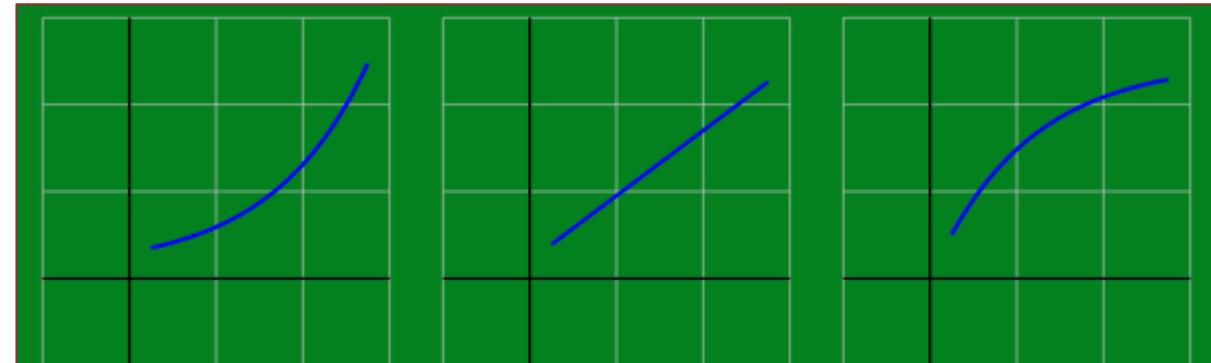
- Próximo do ponto A (ponto em que o gráfico está subindo lentamente), o valor de $f'(x)$ é positivo e relativamente próximo de zero;
- Em contrapartida, perto do ponto B (ponto em que o gráfico cai rapidamente), a derivada $f'(x)$ é negativa e relativamente alta em valor absoluto.



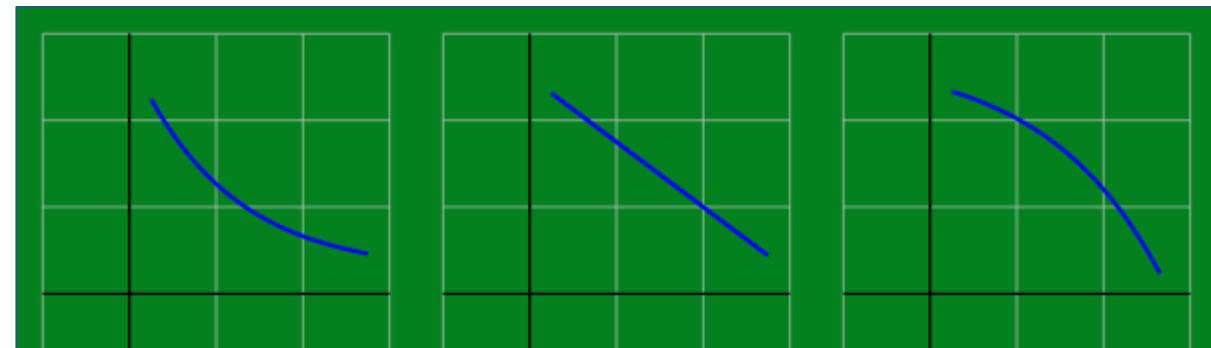
SEGUNDA DERIVADA

A segunda derivada (ou derivada de segunda ordem) é a derivada da função derivada. Assim como a primeira derivada mede a taxa de variação da função original, a segunda derivada mede a taxa de variação da primeira derivada.

A segunda derivada nos ajudará a entender **como** a taxa de variação da função original está mudando.

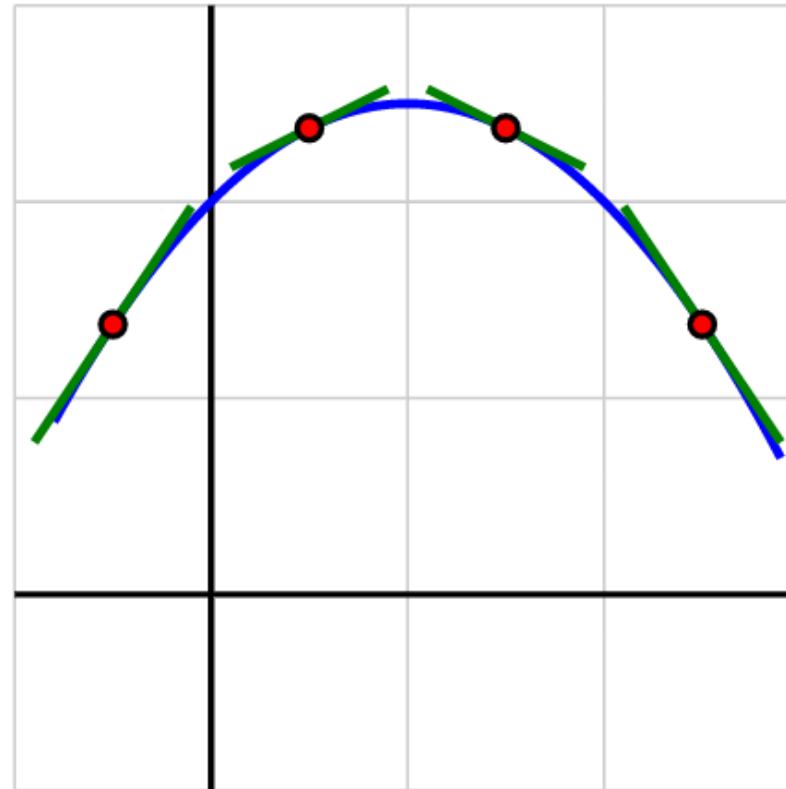
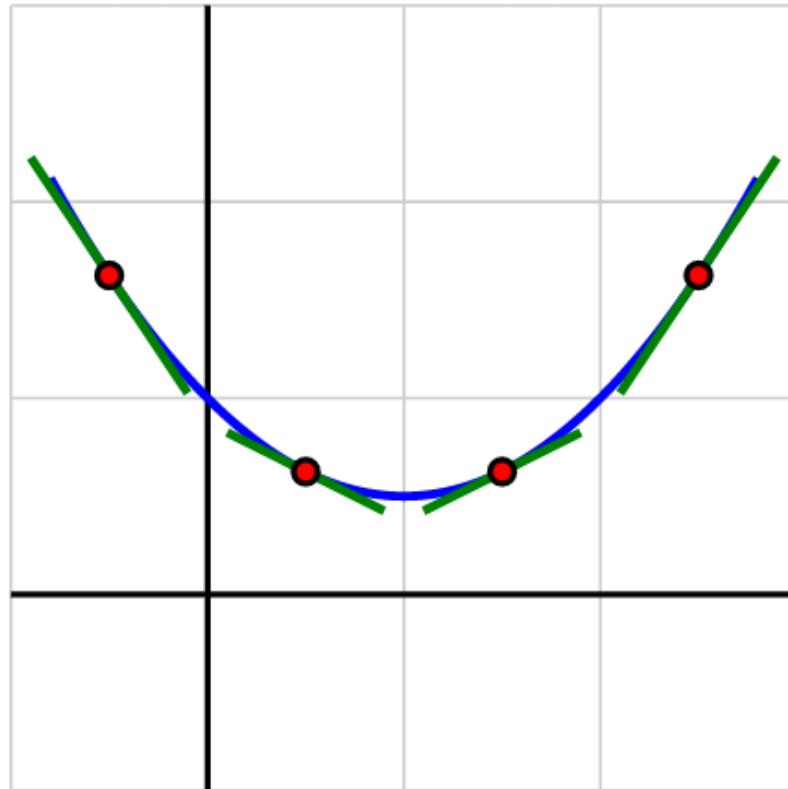


três funções que estão todas aumentando, de maneiras diferentes



três funções que estão todas diminuindo, de maneiras diferentes

CONCAVIDADE



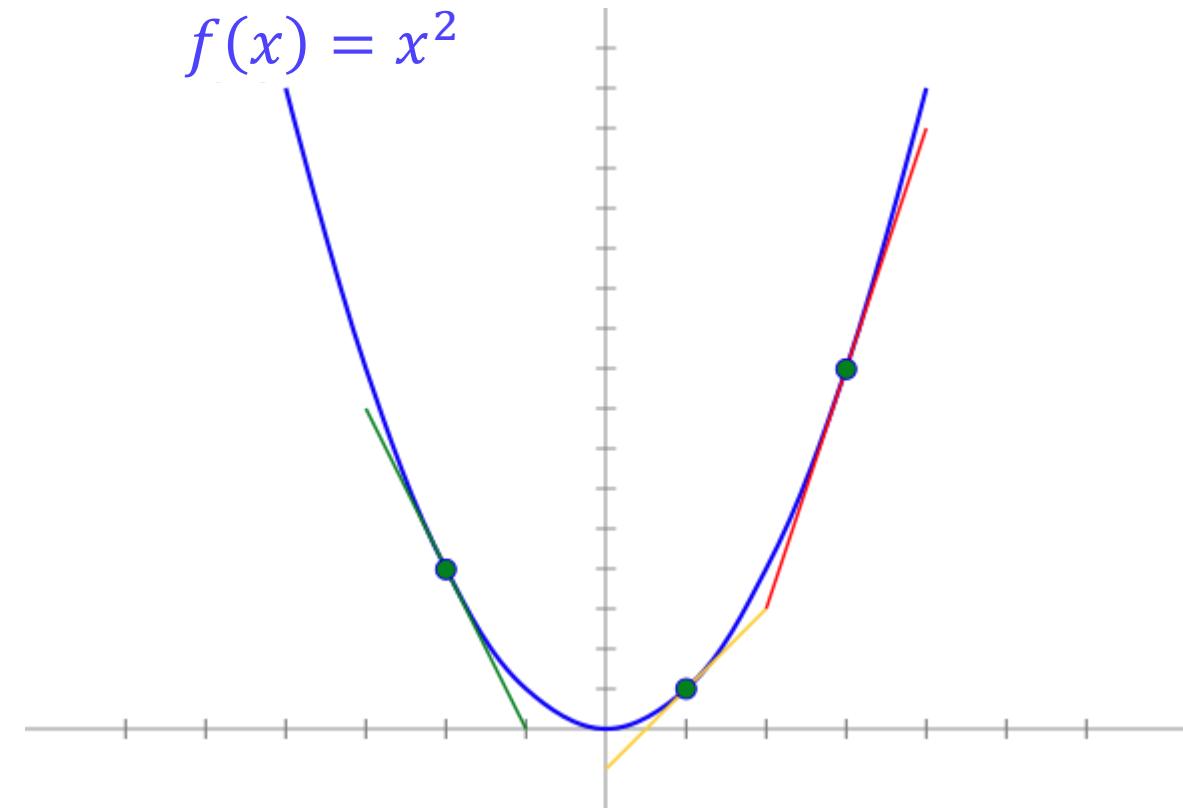
Em um intervalo onde o gráfico de $f(x)$ é côncavo para cima, $f'(x)$ é crescente e $f''(x)$ é positivo. Da mesma forma, em um intervalo onde o gráfico de $f(x)$ é côncavo para baixo, $f'(x)$ é decrescente e $f''(x)$ é negativo.

POR EXEMPLO

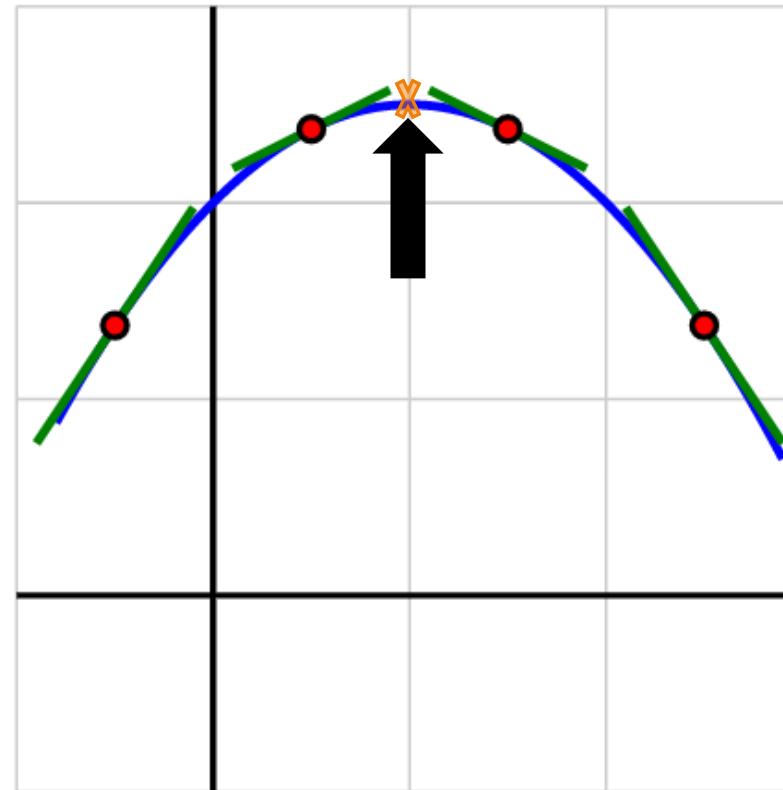
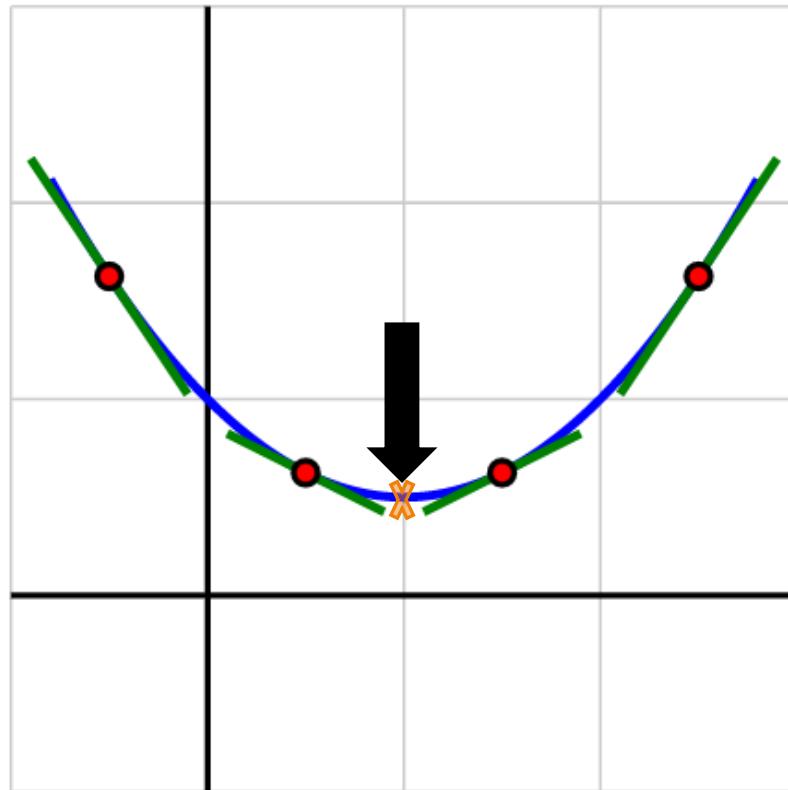
A segunda derivada da função $f(x) = x^2$ é

$$f''(x) = 2$$

$f(x) = x^2$ é côncavo para cima;
 $f'(x) = 2x$ é crescente;
 $f''(x) = 2$ é positivo!



MINIMO E MÁXIMO



Qual é o valor da derivada no ponto X?

DERIVADA PARCIAL

$$f(x, y) = 2x^2 + y^2$$

Portanto,

Símbolo de derivada parcial

$$\boxed{\frac{\partial f(x, y)}{\partial x}} = 2x \quad \frac{\partial f(x, y)}{\partial y} = 2y$$



GRADIENTE

Para otimizar uma solução ou minimizar um erro, precisamos saber como diferenciar funções objetivo e erros , que são expressas com matrizes e vetores.

INTUIÇÃO INICIAL DE GRADIENTE



Subir a
colina de
olhos
vendados,
dando
passos na
direção
3D mais
íngreme

AFINAL, O QUE É GRADIENTE?

"Um gradiente mede o quanto a saída de uma função muda se você alterar um pouco as entradas."

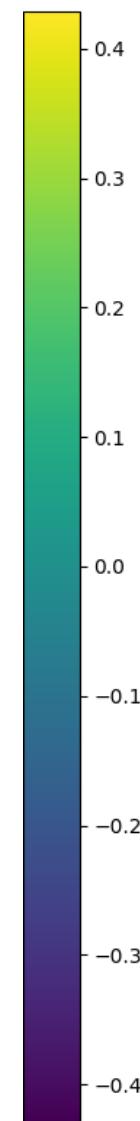
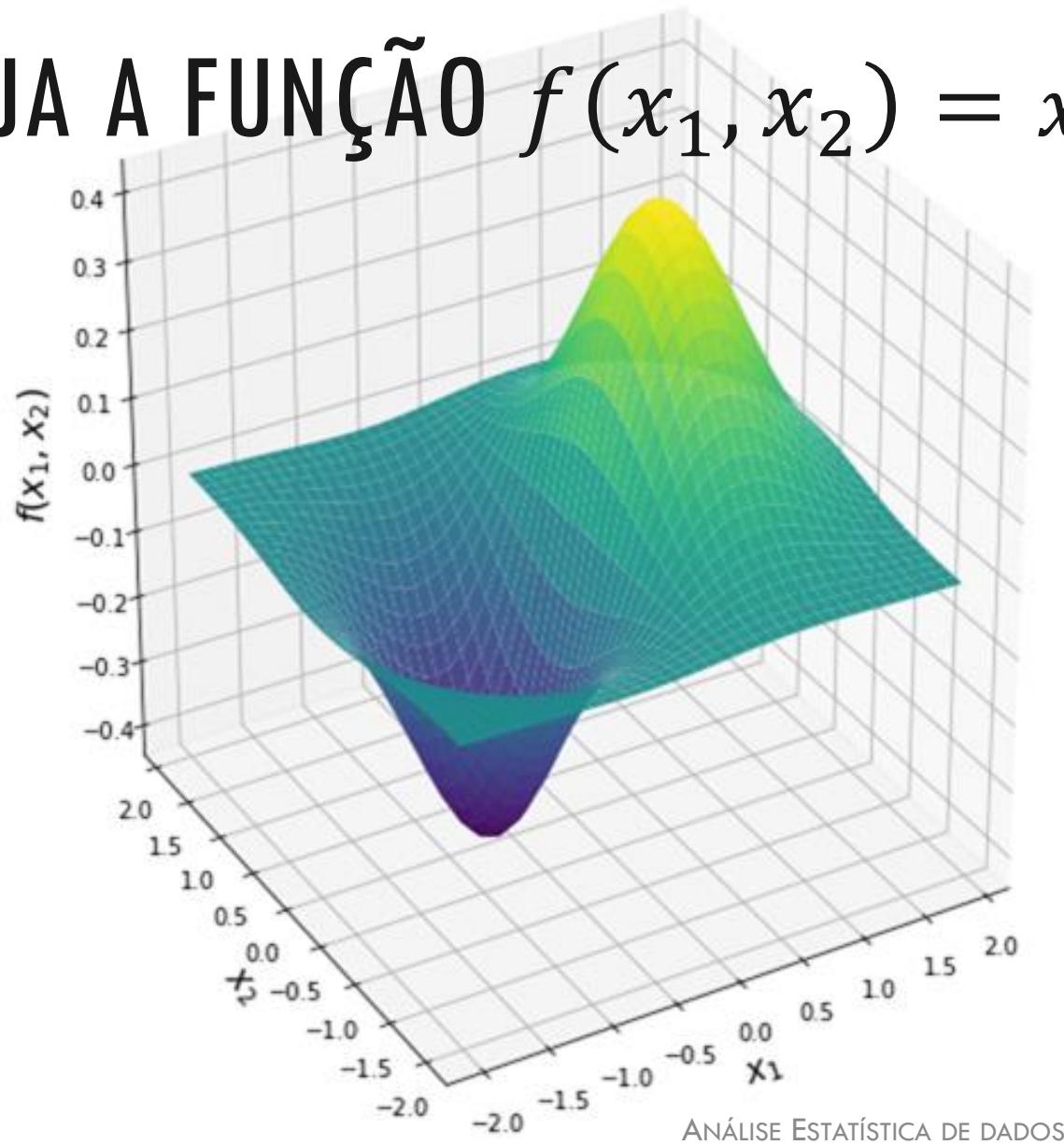
E o bom é que o gradiente é exatamente a mesma coisa que a derivada!

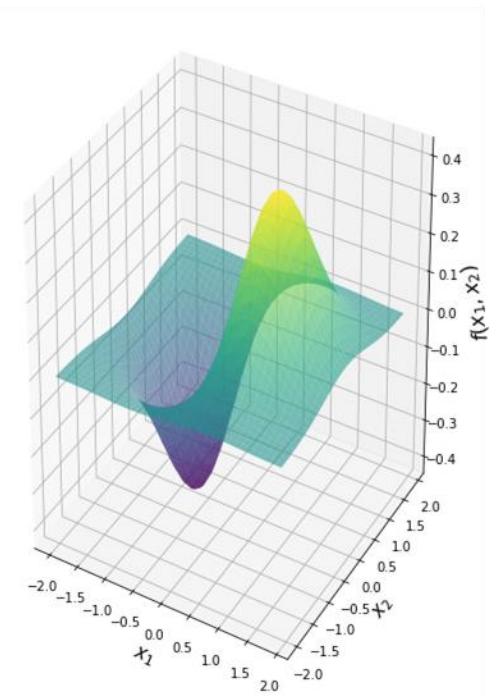
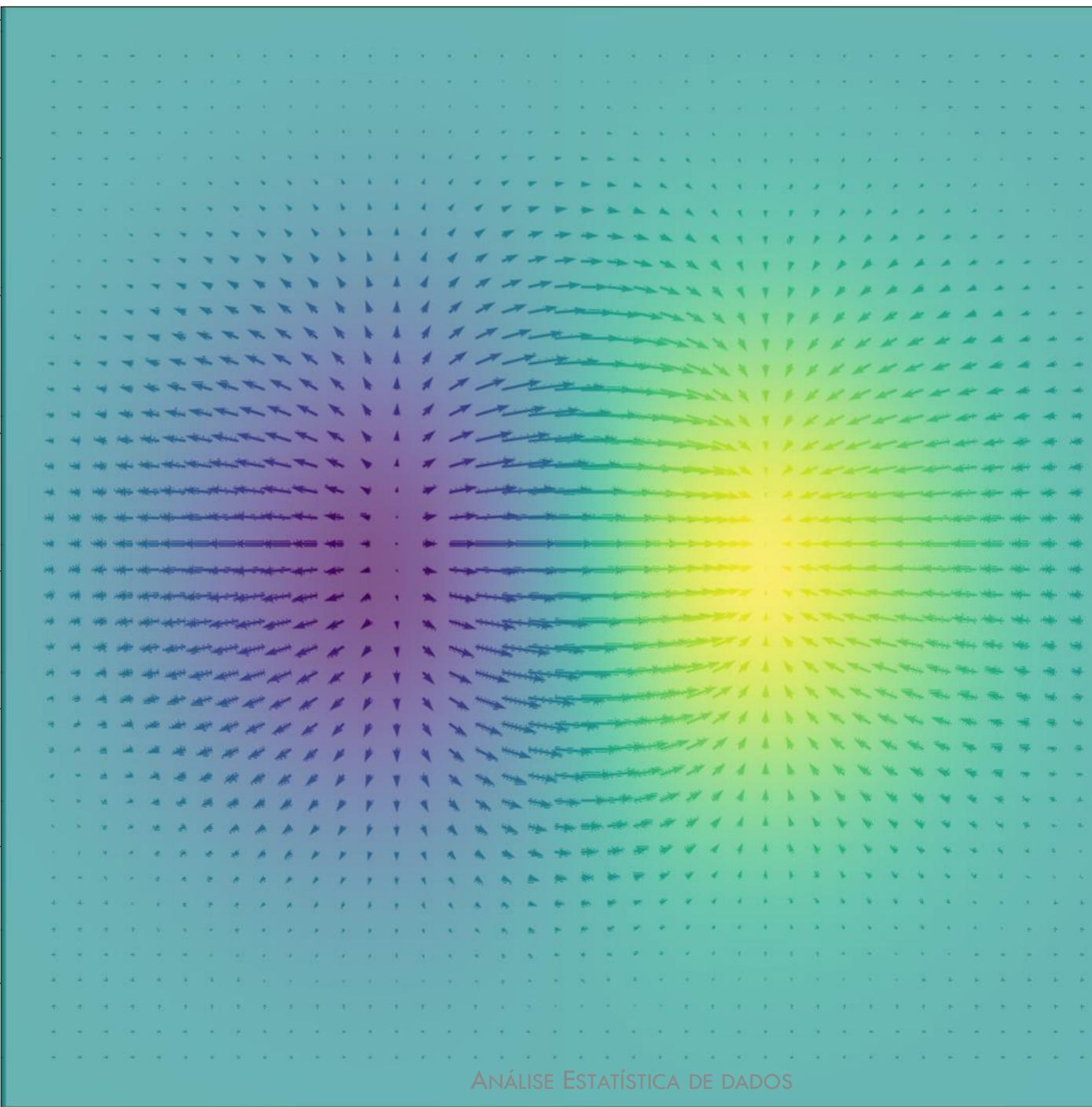
Com uma exceção, o Gradiente é uma função de valor vetorial que armazena derivadas parciais. Em outras palavras, o gradiente é um vetor e cada um de seus componentes é uma derivada parcial em relação a uma variável específica.

No caso de uma função univariada, é simplesmente a primeira derivada em um ponto selecionado.

No caso de uma função multivariada, é um vetor de derivadas em cada direção (ao longo dos eixos das variáveis).

VEJA A FUNÇÃO $f(x_1, x_2) = x_1 e^{x_1^2 - x_2^2}$





GRADIENTE DA FUNÇÃO

$$f(x_1, x_2) = x_1^2 + x_2^2$$

$$\frac{\partial f(x_1, x_2)}{\partial x_1} = 2x_1 \quad \frac{\partial f(x_1, x_2)}{\partial x_2} = 2x_2$$

Então o gradiente é o seguinte vetor:

$$\nabla f(x_1, x_2) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix}$$

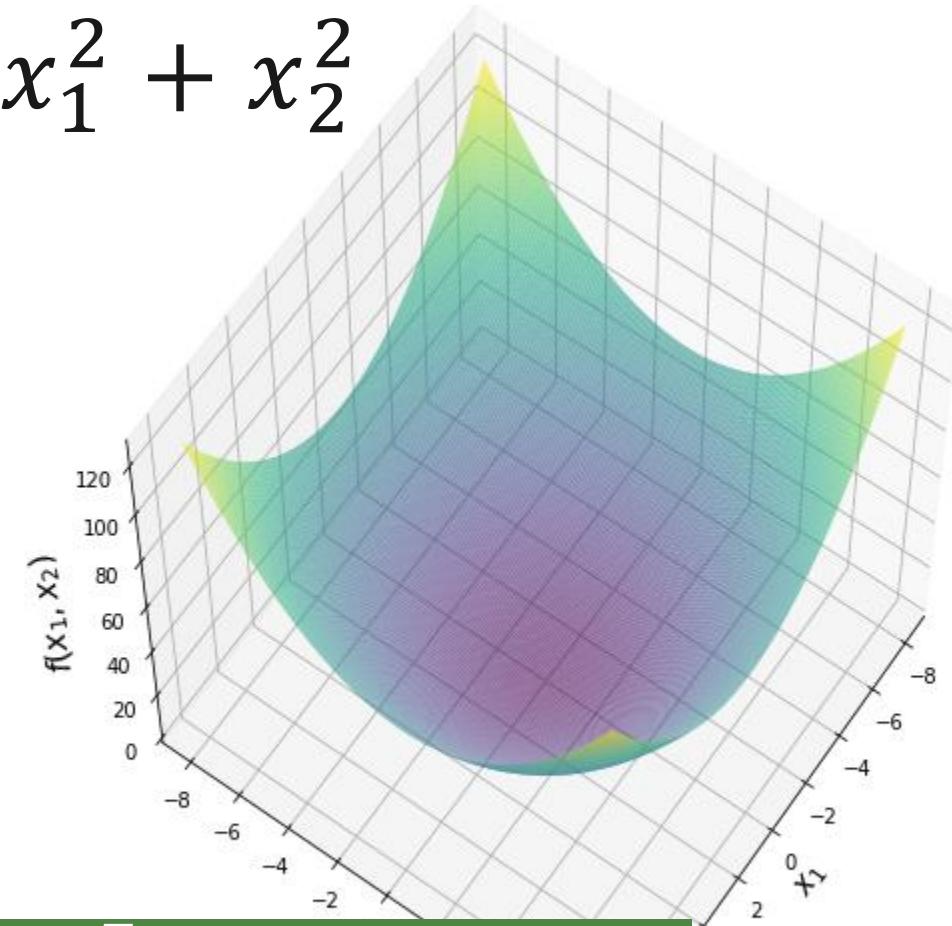
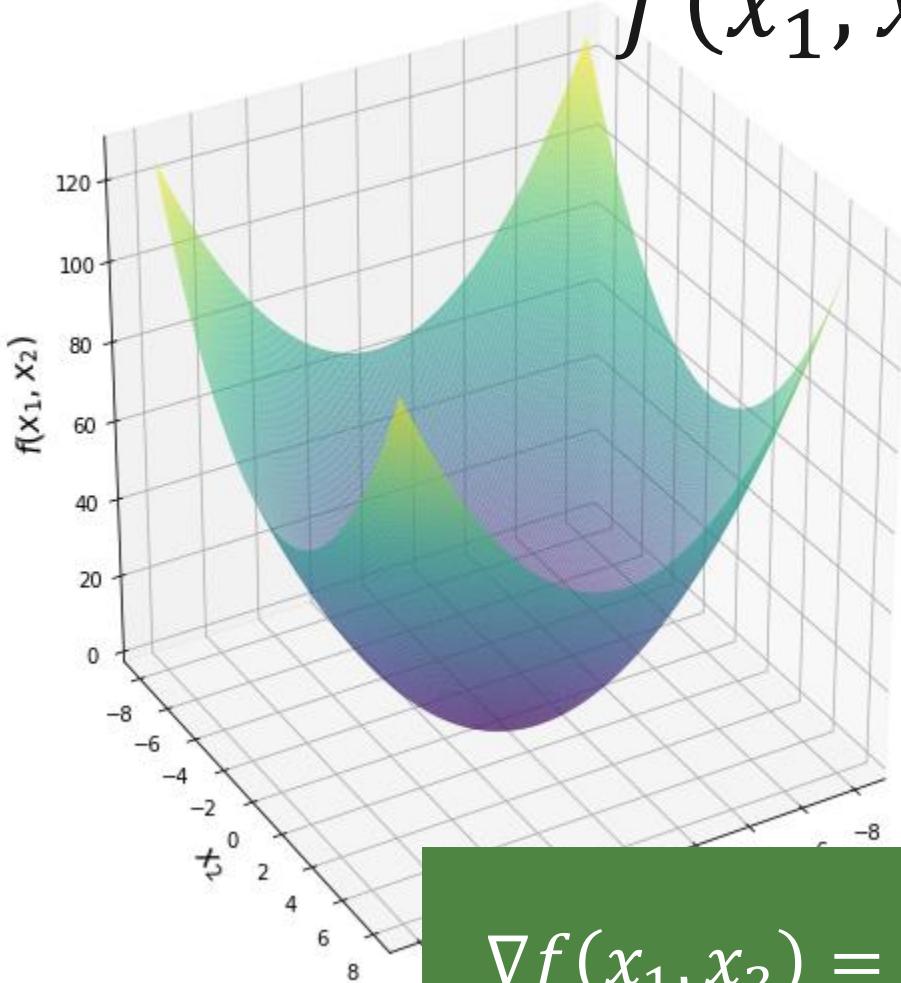
Observe que cada componente indica qual é a direção de subida mais íngreme para cada uma das variáveis da função. Em outras palavras, **o gradiente aponta para a direção em que a função aumenta mais.**

Generalizando,

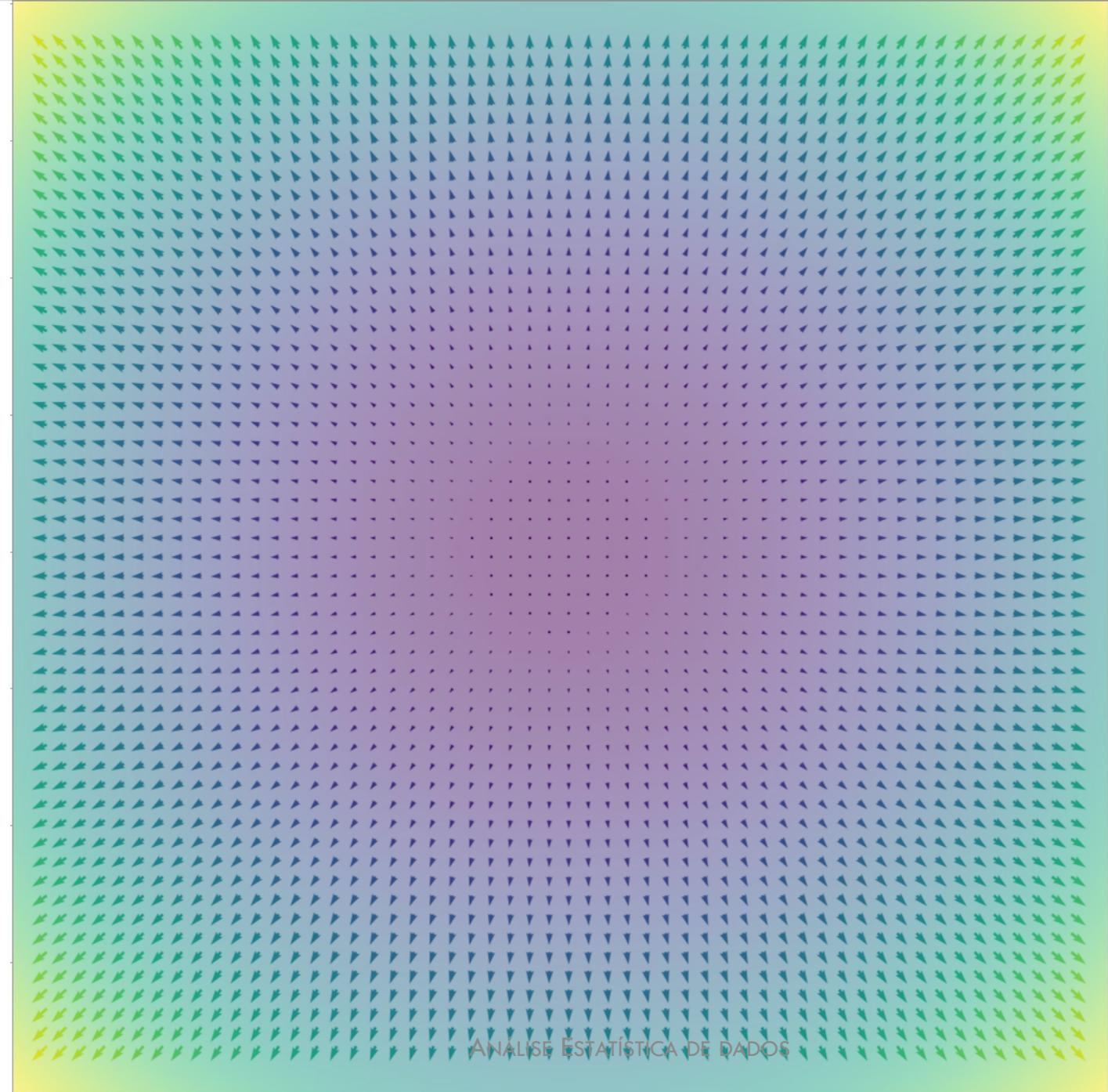
$$\nabla f(x_1, x_2, \dots, x_n) = \left[\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \dots \quad \frac{\partial f}{\partial x_n} \right]^T$$

POR EXEMPLO,

$$f(x_1, x_2) = x_1^2 + x_2^2$$



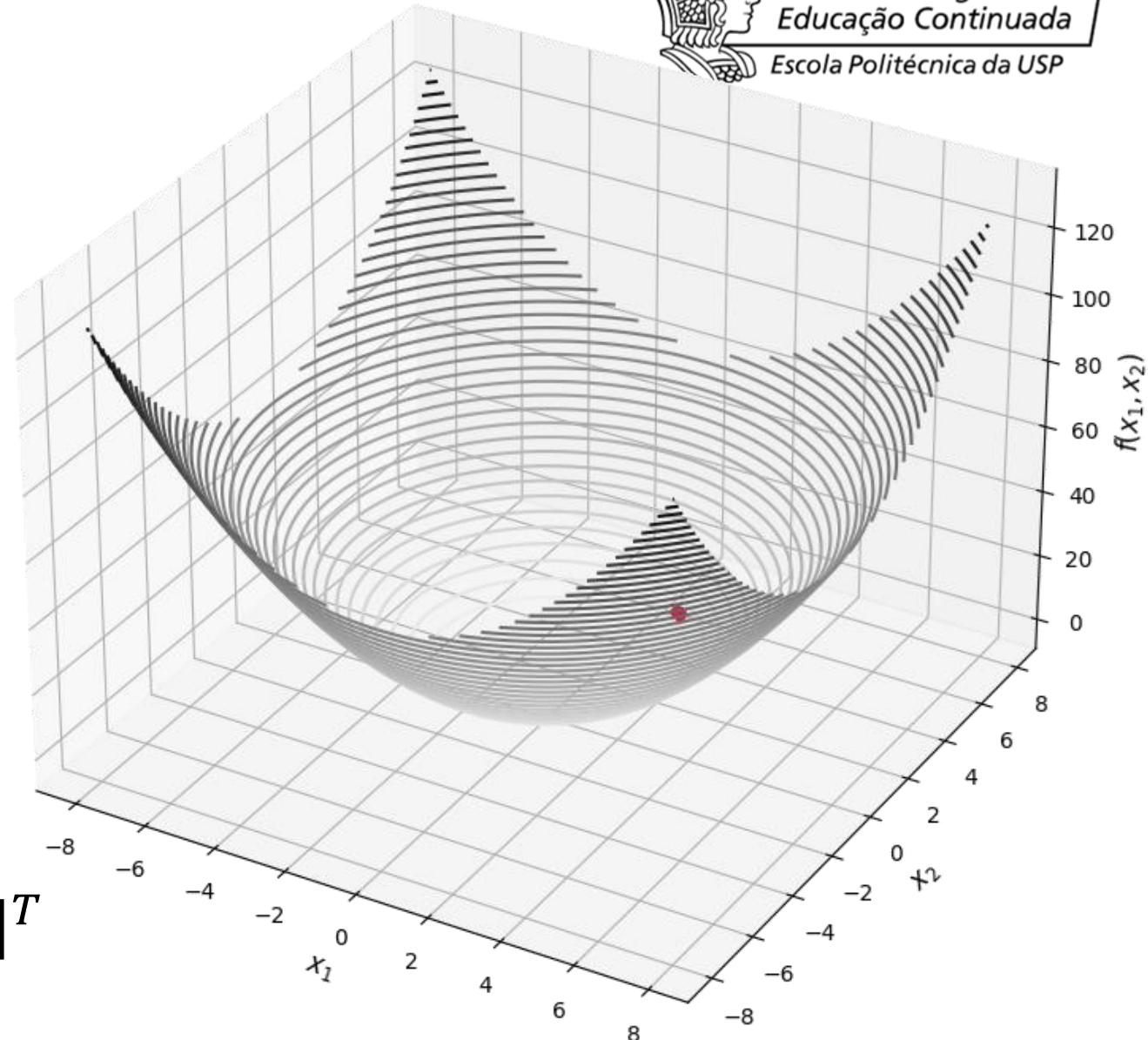
$$\nabla f(x_1, x_2) = \left[\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \right]^T = [2x_1 \quad 2x_2]^T$$



Se quisermos encontrar a direção a ser movida para aumentar nossa função o mais rápido, inserimos nossas coordenadas atuais (como, por exemplo, 2,3) no gradiente:

$$\nabla f(x_1, x_2) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix}$$

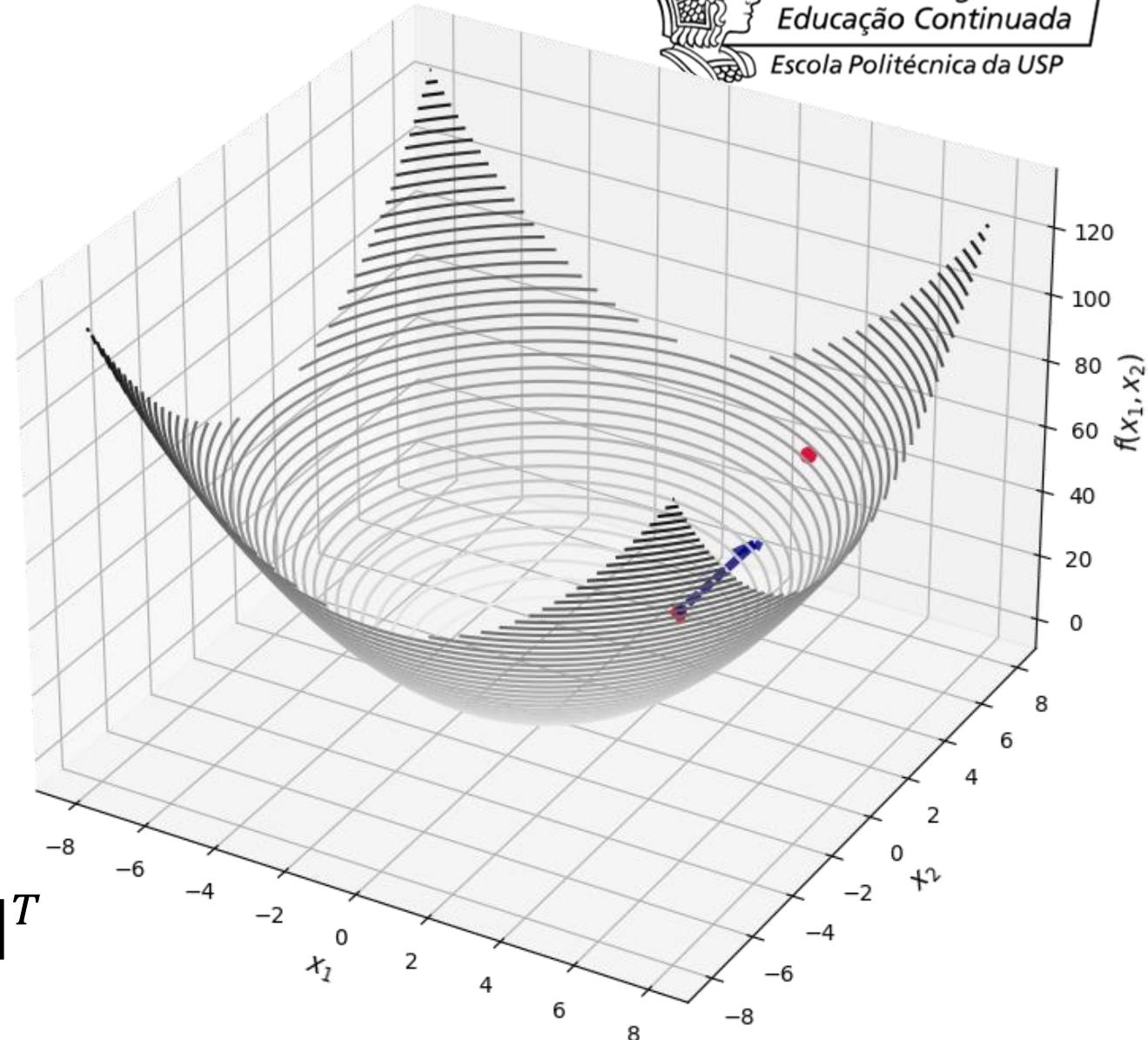
$$\nabla f(x_1, x_2) = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix}^T = [4 \quad 6]^T$$



Se quisermos encontrar a direção a ser movida para aumentar nossa função o mais rápido, inserimos nossas coordenadas atuais (como, por exemplo, 2,3) no gradiente:

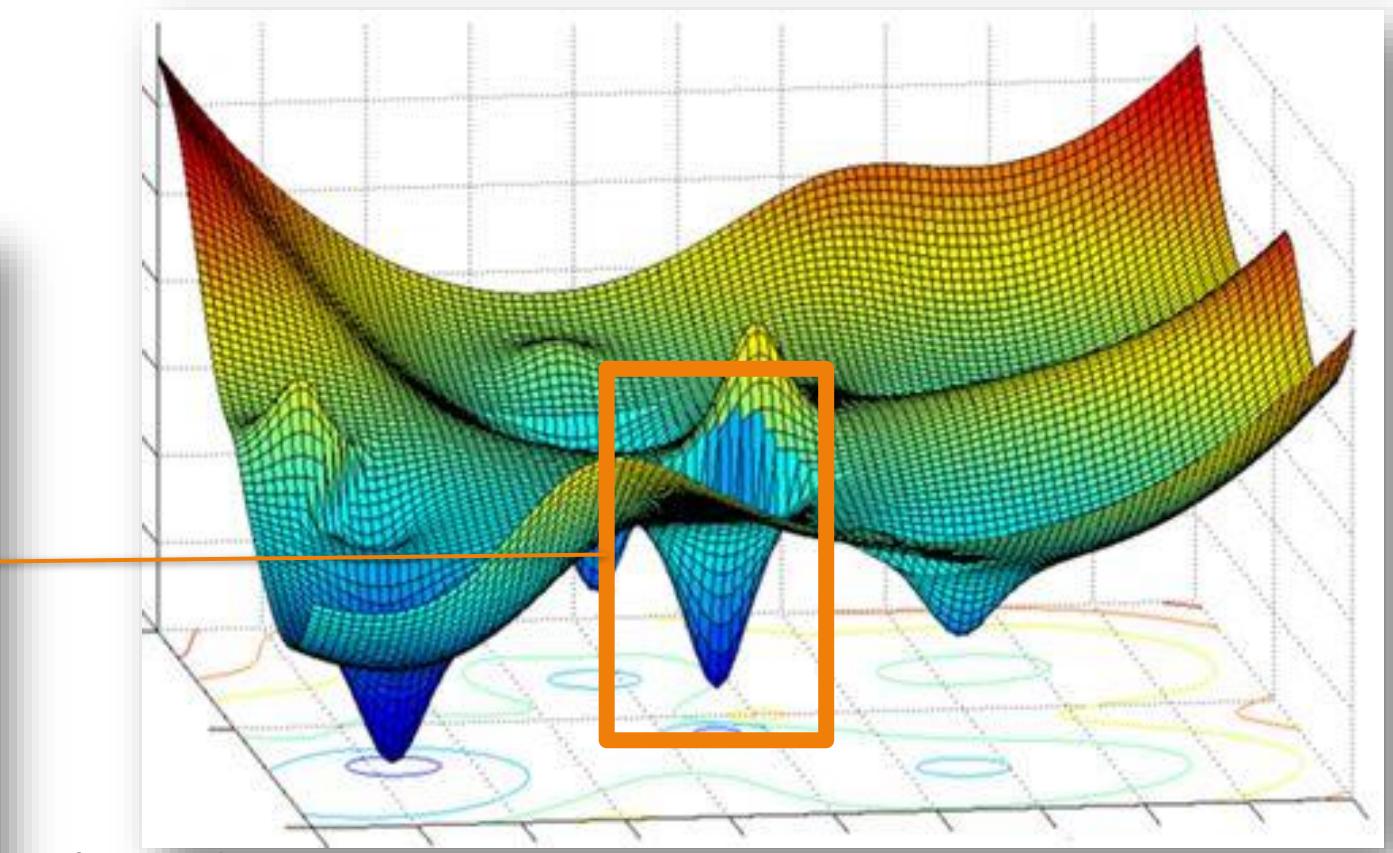
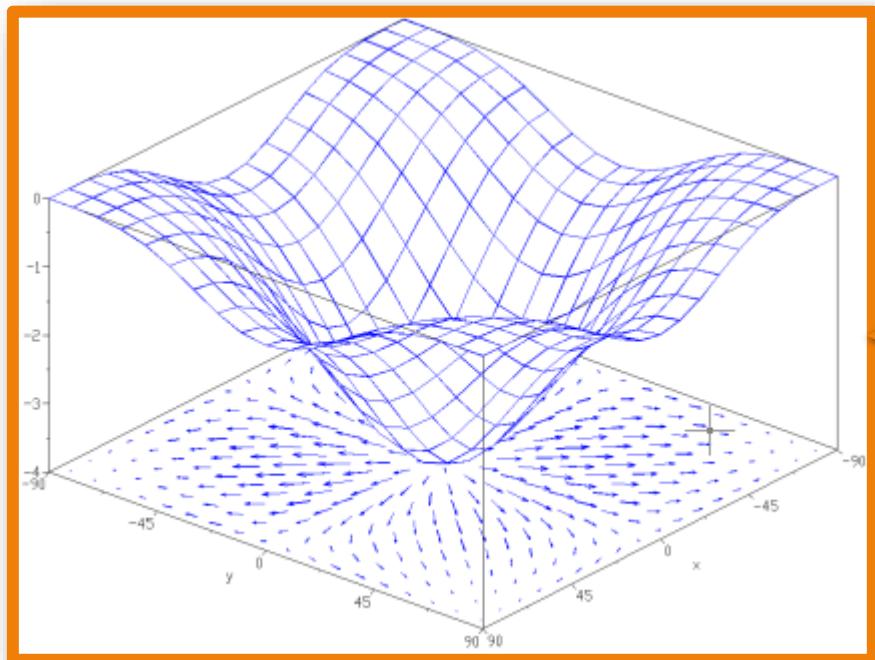
$$\nabla f(x_1, x_2) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix}$$

$$\nabla f(x_1, x_2) = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix}^T = [4 \quad 6]^T$$



FUNÇÕES MAIS COMPLEXAS

Muitas funções são bastante complexas!

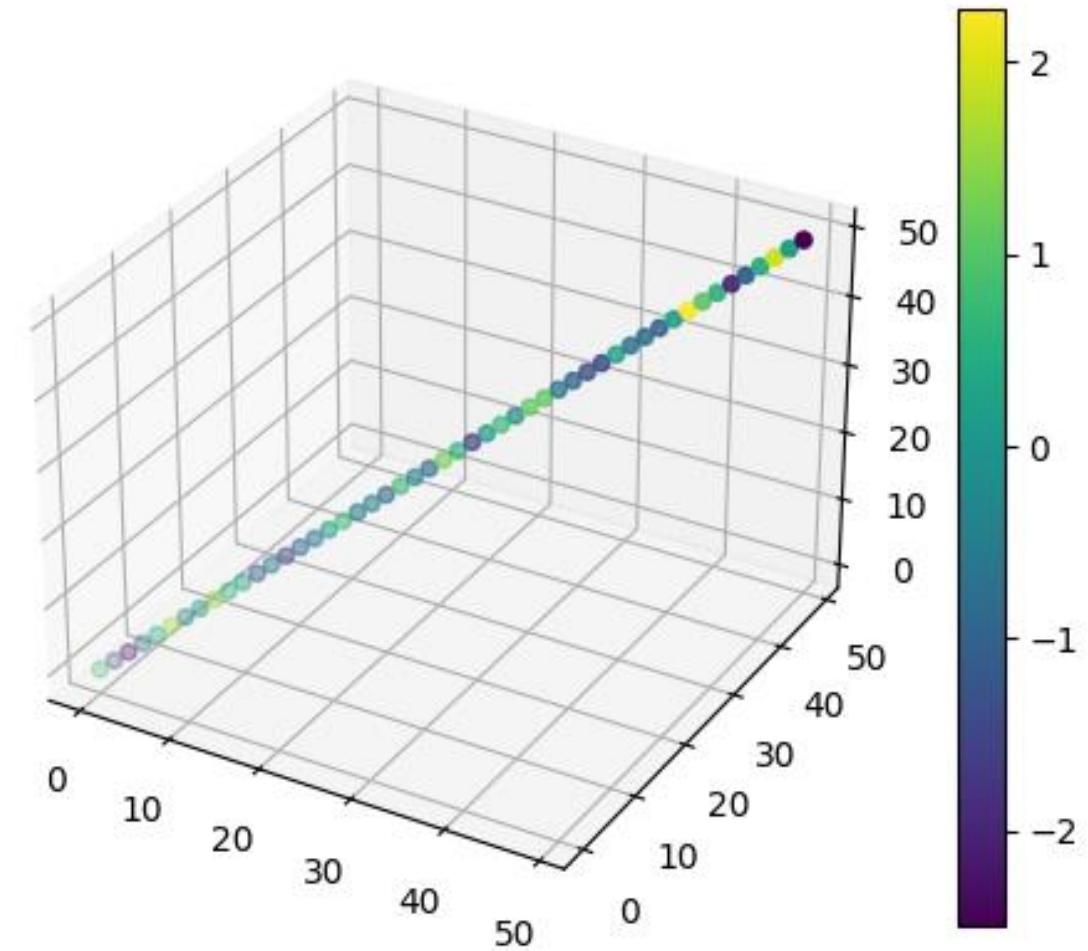


FUNÇÕES COM MUITAS ENTRADAS

Muitas funções têm múltiplas entradas, e cada entrada pode ser considerada uma direção diferente no espaço.

A figura ao lado tem 4 dimensões...

O que acontece se você tiver um problema com 12 entradas?





Para essa função, há uma taxa de variação diferente para cada direção.

Mas, calma! Você não precisa visualizar 12 dimensões!!!!

Então...

GENERALIZANDO

O gradiente é um vetor $n \times 1$ de derivadas parciais,

$$\nabla_x f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_m} \end{bmatrix}$$

Como visualizamos em
nossos exemplos!

JACOBIANO

Suponha que tenhamos m funções $f_i(x_1, x_2, \dots, x_n)$ de dimensão n . Então, o gradiente de f_i com respeito a x_1, x_2, \dots, x_n

$$\nabla f_i(x_1, x_2, \dots, x_n) = \left[\frac{\partial f_i}{\partial x_1} \quad \frac{\partial f_i}{\partial x_2} \quad \dots \quad \frac{\partial f_i}{\partial x_n} \right]^T$$

A **Jacobiana** é uma matriz $m \times n$ de derivadas parciais,

$$J = [\nabla f_1 \quad \nabla f_2 \quad \dots \quad \nabla f_m]^T = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

$$(J)_{ij} = \frac{\partial f_i}{\partial x_j}$$

ALGUMAS REGRAS GERAIS

Dado $\mathbf{A}\mathbf{x} - \mathbf{b} = \mathbf{0}$

Podemos calcular o Jacobiano,

$$J_x(\mathbf{A}\mathbf{x} - \mathbf{b}) = J_x(\mathbf{A}\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1(\mathbf{A}\mathbf{x})}{\partial x_1} & \frac{\partial f_1(\mathbf{A}\mathbf{x})}{\partial x_2} \\ \frac{\partial f_2(\mathbf{A}\mathbf{x})}{\partial x_1} & \frac{\partial f_2(\mathbf{A}\mathbf{x})}{\partial x_2} \\ \vdots & \vdots \\ \frac{\partial f_3(\mathbf{A}\mathbf{x})}{\partial x_1} & \frac{\partial f_3(\mathbf{A}\mathbf{x})}{\partial x_2} \end{bmatrix} = \mathbf{A}$$

Extrapolando...

$$J_x(\mathbf{x}^T \mathbf{A} \mathbf{x}) = 2\mathbf{A}\mathbf{x}$$

$$J_x(\mathbf{v}^T \mathbf{A} \mathbf{x}) = 2\mathbf{A}\mathbf{v}$$

Por exemplo, $\mathbf{A}\mathbf{x}$ pode ser escrito como,

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \begin{bmatrix} a_{11}x_1 + a_{12}x_2 \\ a_{21}x_1 + a_{22}x_2 \\ a_{31}x_1 + a_{32}x_2 \end{bmatrix} \begin{matrix} f_1 \\ f_2 \\ f_3 \end{matrix}$$

$$J_x(\mathbf{A}\mathbf{x}) = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}$$

HESSIANA

E as derivadas de segunda ordem, ié, a derivada da derivada?

A **Hessiana** é uma **matriz** que organiza todas as derivadas parciais de segunda ordem de uma função. Suponha que $f: \mathbb{R}^n \rightarrow R$. Então a matriz hessiana em relação a x , escrita $\nabla_x^2 f(x) = H$, é uma matriz $n \times n$ simétrica de derivadas parciais,

$$\nabla_x^2 f(x) \in \mathbb{R}^{nxn} = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \dots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}$$

Cada linha representa a alteração do gradiente em uma determinada direção.

$$(\nabla_x^2 f(x))_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$$

EXEMPLO

$$\mathbf{v} = \begin{bmatrix} f_1(x, y) \\ f_2(x, y) \end{bmatrix} = \begin{bmatrix} x^4 + 2x^2y^2 \\ 5y^2 - 4xy + 1 \end{bmatrix}$$

$$\frac{\partial f_1}{\partial x} = 4x^3 + 4xy^2$$

$$\frac{\partial f_1}{\partial y} = 4x^2y$$

$$\frac{\partial f_2}{\partial x} = -4y$$

$$\frac{\partial f_2}{\partial y} = 10y - 4x$$

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}$$

$$J = \begin{bmatrix} 4x^3 + 4xy^2 & 4x^2y \\ -4y & 10y - 4x \end{bmatrix}$$



PROBLEMA DE OTIMIZAÇÃO

Que é ótimo?

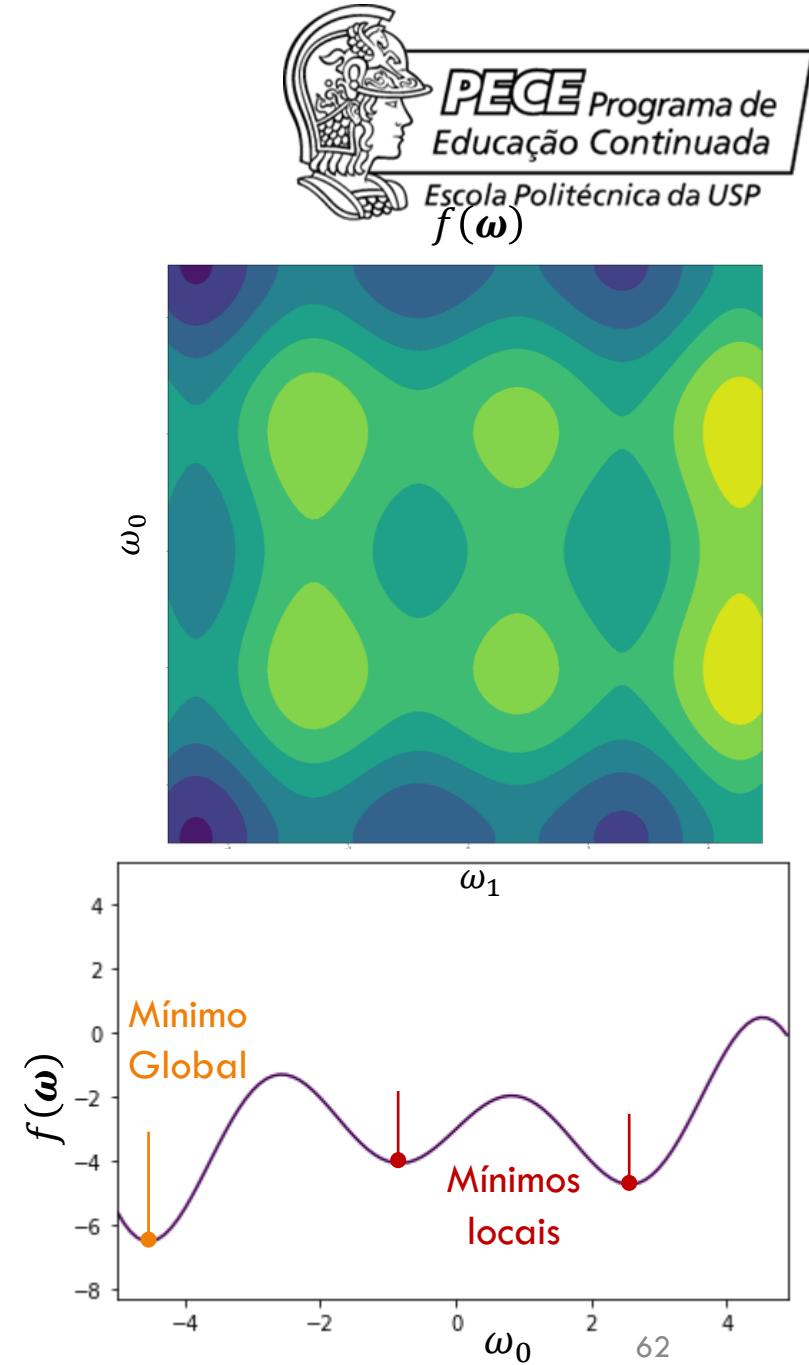
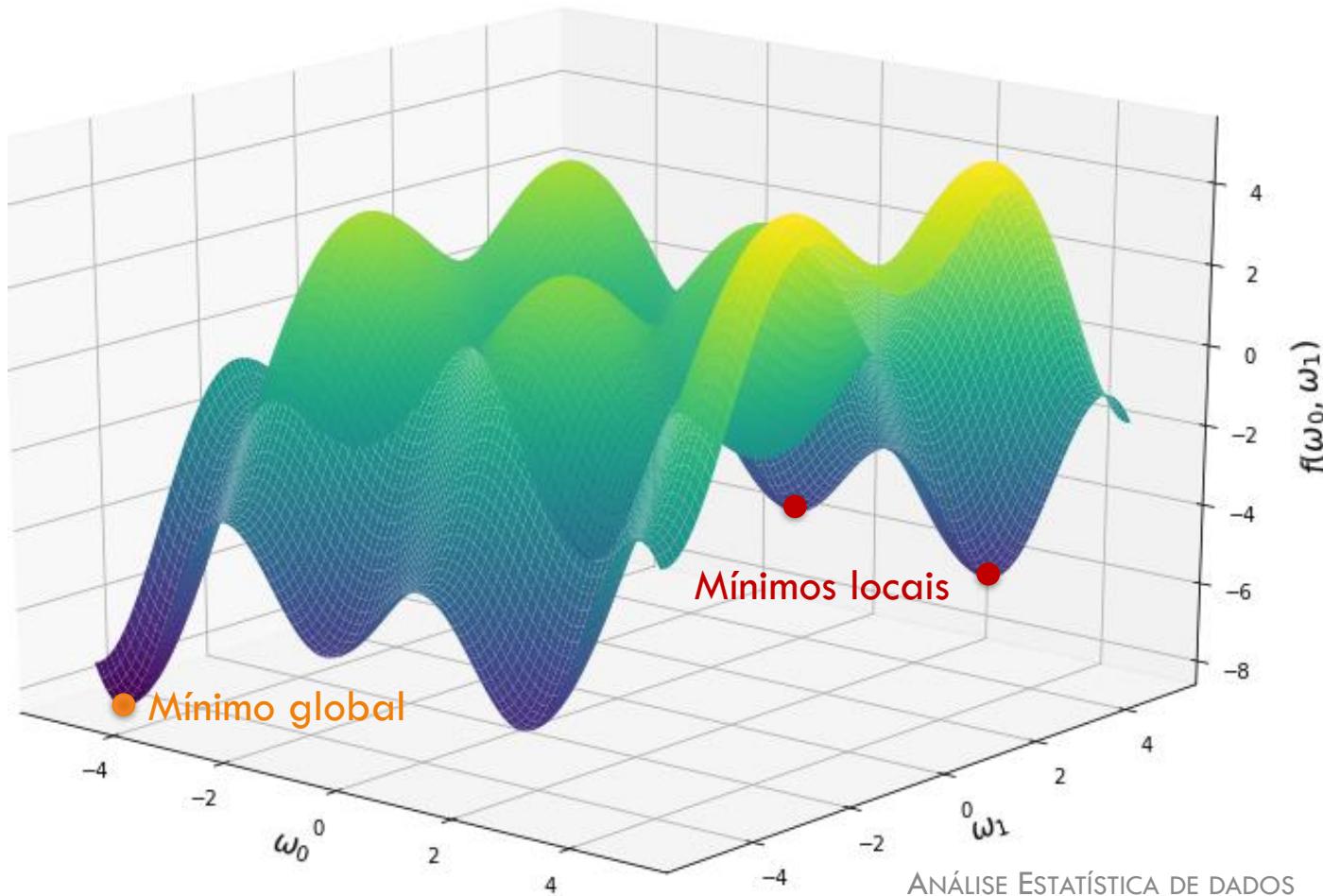
OTIMIZAÇÃO EM ENGENHARIA

- Olá, tudo bem?
- Sim, todo ótimo!
- Oh, sinto muito. (Agora as coisas só podem piorar).

Otimização:

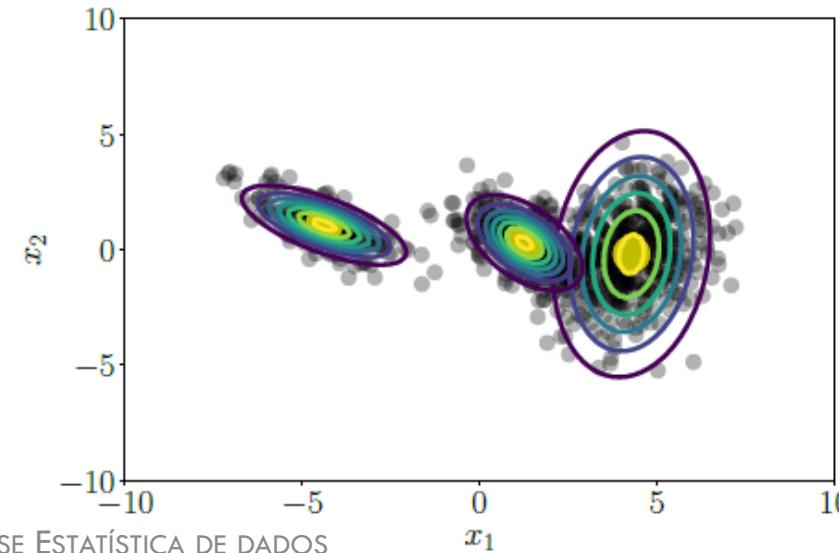
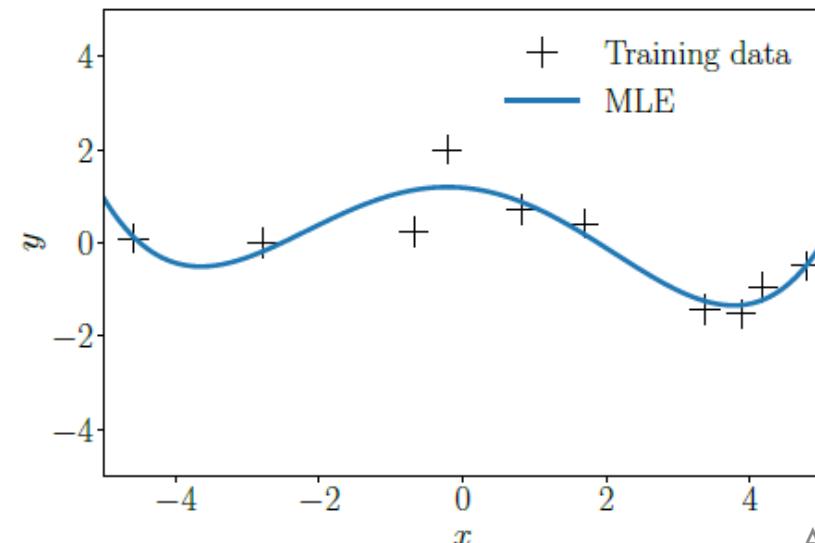
Processo para encontrar **a melhor solução possível** para um problema, considerando os **objetivos** da solução e as **restrições** impostas ao problema.

COMO RESOLVER O PROBLEMA DE OTIMIZAÇÃO???

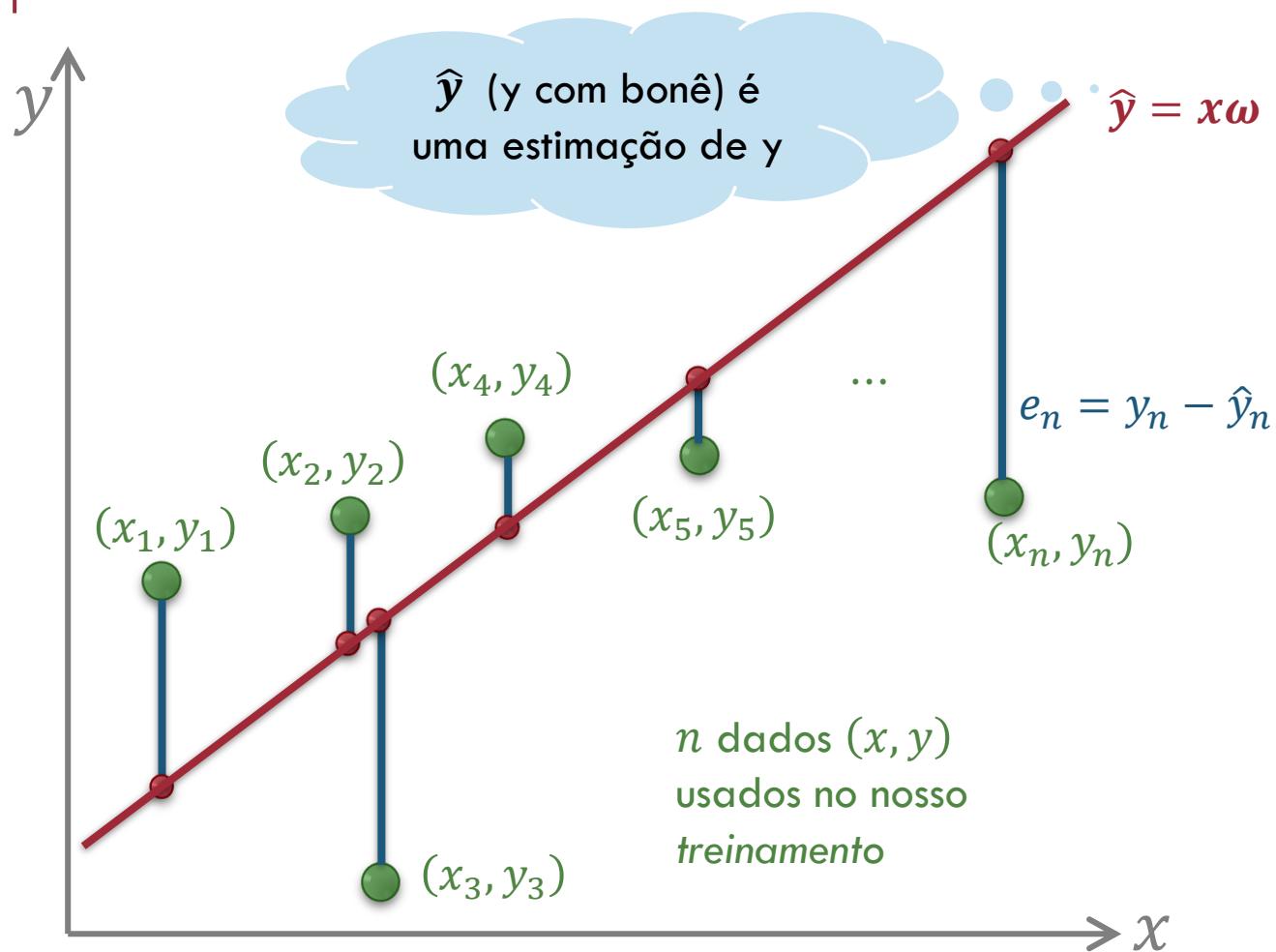


MOTIVAÇÃO

Muitos algoritmos em aprendizado de máquina otimizam uma função objetivo em relação a um conjunto de parâmetros de modelo desejados que controlam quanto bem um modelo explica os dados: **Encontrar bons parâmetros pode ser formulado como um problema de otimização.**



POR EXEMPLO, PROBLEMA DE REGRESSÃO



$$\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \omega_0 \\ \omega_1 \end{bmatrix} = \begin{bmatrix} \omega_0 + \omega_1 x_1 \\ \omega_0 + \omega_1 x_2 \\ \vdots \\ \omega_0 + \omega_1 x_n \end{bmatrix}$$

Vetor $n \times 1$
que contém
as previsões
pontuais

$$e(\omega_0, \omega_1) = y - \mathbf{x}\boldsymbol{\omega} = y - \hat{y}$$

$$MSE = \frac{1}{n} \sum_{i=1}^n e_i^2(\boldsymbol{\omega})$$

(Última aula falamos que MSE é o produto interno $e \cdot e$, lebram-se?)

VEJA QUE...

$$e(\beta_0, \beta_1) = \mathbf{y} - \mathbf{x}\boldsymbol{\omega} = \mathbf{y} - \hat{\mathbf{y}}$$

$$\mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - \mathbf{x}\boldsymbol{\omega} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} - \begin{bmatrix} \omega_0 + \omega_1 x_1 \\ \omega_0 + \omega_1 x_2 \\ \vdots \\ \omega_0 + \omega_1 x_n \end{bmatrix} = \begin{bmatrix} y_1 - \omega_0 - \omega_1 x_1 \\ y_2 - \omega_0 - \omega_1 x_2 \\ \vdots \\ y_n - \omega_0 - \omega_1 x_n \end{bmatrix}$$

Cuidado! Os índices $1, 2, \dots, n$ se referem a diferentes valores do atributo x .
 Aqui nossa função depende de ω_0, ω_1

ERRO MÍNIMO QUADRADO

Suponha uma matriz $X \in \mathbb{R}^{m \times n}$ e um vetor $\omega \in \mathbb{R}^m$. Em muitas situações, não conseguiremos encontrar um vetor $y \in \mathbb{R}^n$, tal que

$$X\omega = y$$

Então, em vez disso, queremos encontrar um vetor ω tal que $A\omega$ seja o mais próximo possível y , medido pelo quadrado da norma,

$$\|X\omega - y\|^2$$

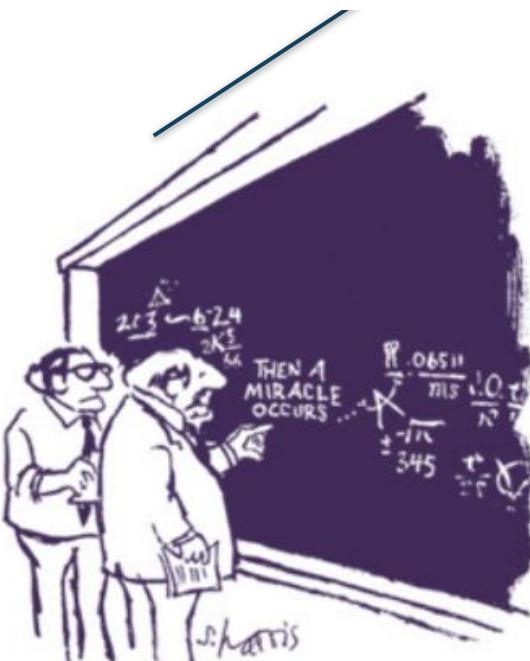
$$\begin{aligned}
 |X\omega - y|^2 &= (X\omega - y)^T(X\omega - y) \\
 &= \omega^T X^T X \omega - (X\omega)^T y - y^T (X\omega) + y^T y \\
 &= \omega^T X^T X \omega - 2y^T X \omega + y^T y
 \end{aligned}$$

$$\begin{aligned}
 (X\omega - y)^T &= [(X\omega)^T - y^T] \\
 (X\omega)^T &= \omega^T X^T
 \end{aligned}$$

(aula anterior, de novo...)

$X\omega$ e y são vetores,
portanto, vale a
seguinte regra de
produto interno,
 $(X\omega)^T y = y^T (X\omega)$

(aula anterior, de
novo...)



"I think you should be more explicit here in step two."

$$\begin{aligned}
 J_{\omega}(\omega^T X^T X \omega - 2y^T X \omega + y^T y) &= \\
 = J_{\omega}(\omega^T X^T X \omega) - J_{\omega}(2y^T X \omega) + J_{\omega}(y^T y) &= \\
 = J_{\omega}(\omega^T X^T X \omega) - J_{\omega}(2y^T X \omega) + J_{\omega}(y^T y) &= \\
 &= 2X^T X \omega - 2X^T y
 \end{aligned}$$

$$\begin{aligned}
 J_x(x^T Ax) &= 2Ax \\
 J_x(v^T Ax) &= 2Av
 \end{aligned}$$

Apareceu uma matriz
inversa aí... $AA^{-1} = I$
(aula anterior, de novo...)

os

$$\begin{aligned}
 2X^T X \omega - 2X^T y &= 0 \\
 \omega &= (X^T X)^{-1} X^T y
 \end{aligned}$$

$$\begin{aligned}
 |X\omega - y|^2 &= (X\omega - y)^T(X\omega - y) \\
 &= \omega^T X^T X \omega - (X\omega)^T y - y^T (X\omega) + y^T y \\
 &= \omega^T X^T X \omega - 2y^T X \omega + y^T y
 \end{aligned}$$

$$(X\omega - y)^T = [(X\omega)^T - y^T]$$

$$(X\omega)^T = \omega^T X^T$$

(aula anterior, de novo...)

$X\omega$ e y são vetores,
portanto, vale a
seguinte regra de
produto interno,
 $(X\omega)^T v = v^T (X\omega)$

(aula anterior, de novo...)

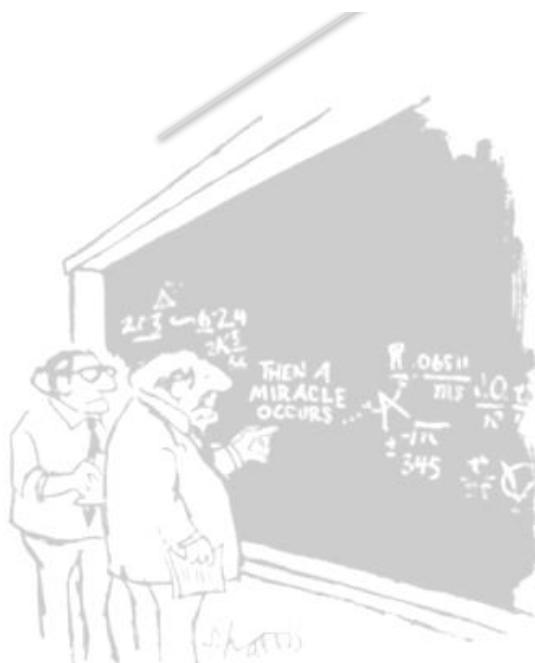
$$\omega = (X^T X)^{-1} X^T y$$

$$\begin{aligned}
 J_\omega(\omega^T X^T X \omega - 2y^T X \omega + y^T y) &= \\
 &= L_\omega(\omega^T X^T X \omega) - L_\omega(2y^T X \omega) + J_\omega(y^T y) \\
 (2y^T X \omega) + J_\omega(y^T y) &= \\
 &= 2X^T X \omega - 2X^T y
 \end{aligned}$$

$$\begin{aligned}
 J_x(x^T A x) &= 2Ax \\
 J_x(v^T A x) &= 2Av
 \end{aligned}$$

$$\begin{aligned}
 2X^T X \omega - 2X^T y &= 0 \\
 \omega &= (X^T X)^{-1} X^T y
 \end{aligned}$$

Apareceu uma matriz
inversa aí... $AA^{-1} = I$
(aula anterior, de novo...) os



"I think you should be more explicit here in
step two."
"THEN A MIRACLE OCCURS"

EQUAÇÃO NORMAL

$$\omega = (X^T X)^{-1} X^T y$$



A equação normal é uma solução de forma fechada para descobrir o valor dos parâmetros ω que minimizam uma determinada função. É chamada de solução de forma fechada no sentido de fornecer o resultado diretamente através da equação.

POR EXEMPLO...

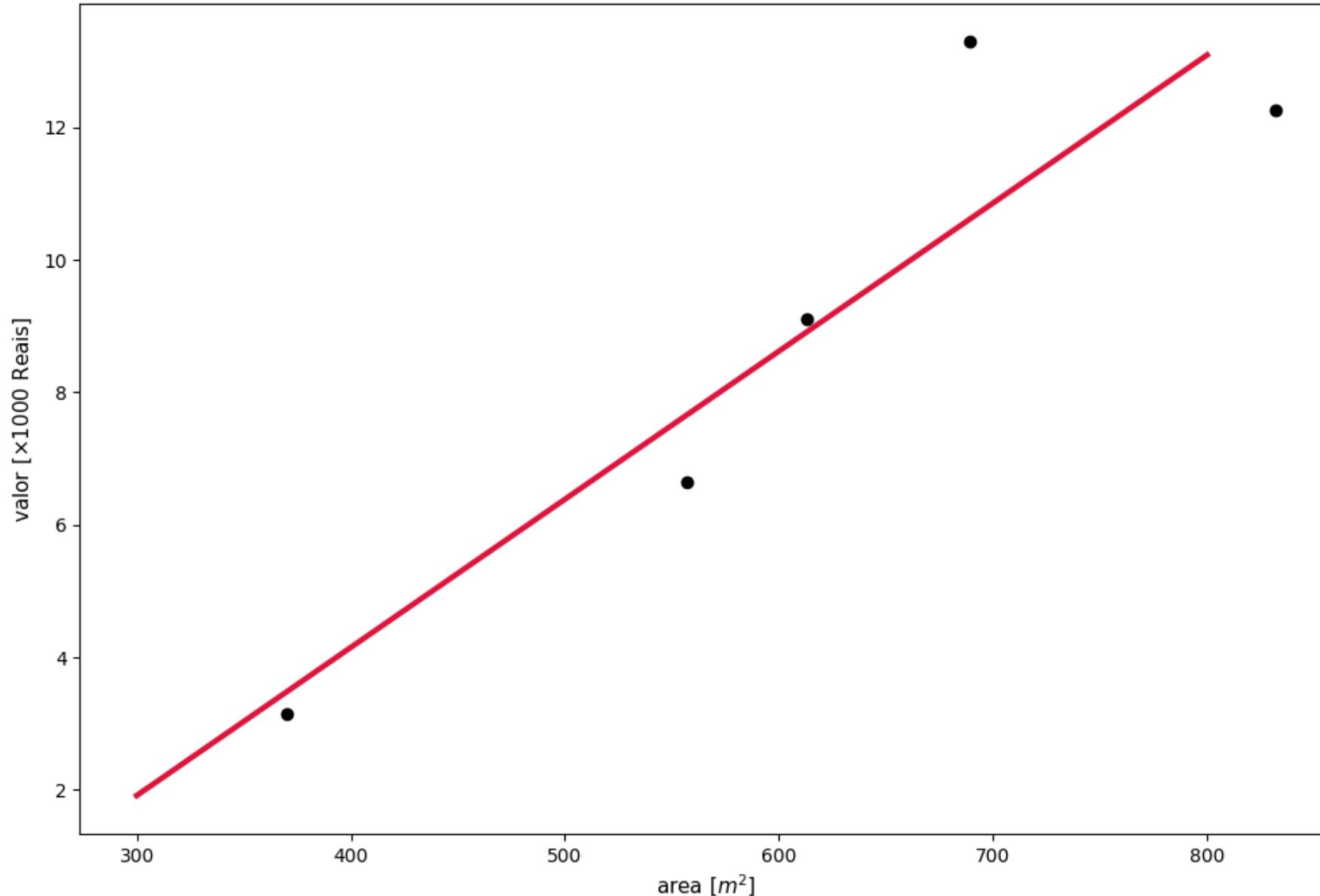


Considerando a tabela abaixo, encontre os parâmetros ótimos para a previsão do preço de um imóvel. Primeiramente, considere apenas um parâmetro – dimensão – depois, aumente e inclua os demais. Compare os resultados.

$$y = \omega_0 x_0 + \omega_1 x_1, \quad x_0 = 1$$

$$\omega = (X^T X)^{-1} X^T y$$

Casa	Dimensão (em m^2)	#quartos	# banheiros	Idade (em anos)	Preço (em milhões de dólares)
01	689	4	2	3	13,300
02	832	4	4	4	12,250
03	613	4	2	2	9,100
04	557	3	2	3	6,650
05	370	2	2	1	3,150



NOTEBOOK

Analise a resposta do Notebook para uma regressão considerando todas as informações.

Imaginando que X é uma matriz com número colunas igual ao número de características m de nosso problema e o número de linhas é o número de dados n que estamos usando para nosso *treinamento*, e pode estar na casa dos milhares, milhões... ($X^T X$) terá a imensa dimensão $n \times n$...

**Você ainda
acha uma
boa ideia
inverter essa
matriz?**





GRADIENTE DESCENDENTE



- Como funciona o gradiente descendente;
- quais tipos são usados hoje;
- suas vantagens e problemas.

PORQUE ESTUDAR GRADIENTE DESCENDENTE?

Porque Gradiente descendente é o coração e a alma da maioria dos algoritmos aprendizado de máquina.

O Gradiente descendente é de longe a estratégia de otimização mais popular usada em aprendizado de máquina e aprendizado profundo no momento.

Ele é usado no treinamento de modelos de dados, pode ser combinado com todos os algoritmos e é fácil de entender e implementar.

Todos que trabalham com aprendizado de máquina devem entender seu conceito.

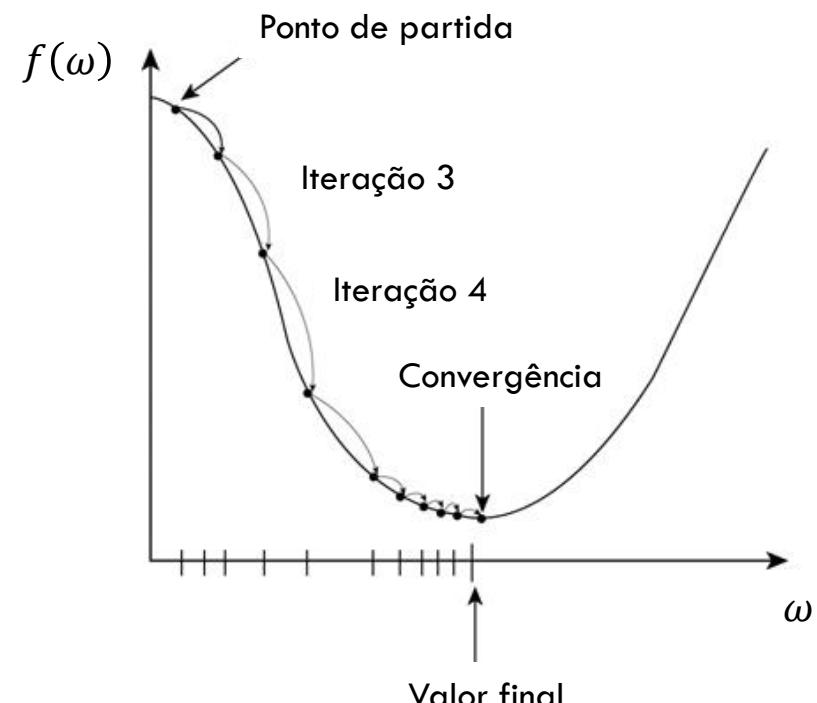
GRADIENTE DESCENDENTE

Na prática, é difícil encontrar um problema sem restrições, com um único mínimo...

Vamos começar do princípio, com um problema com uma única variável ω

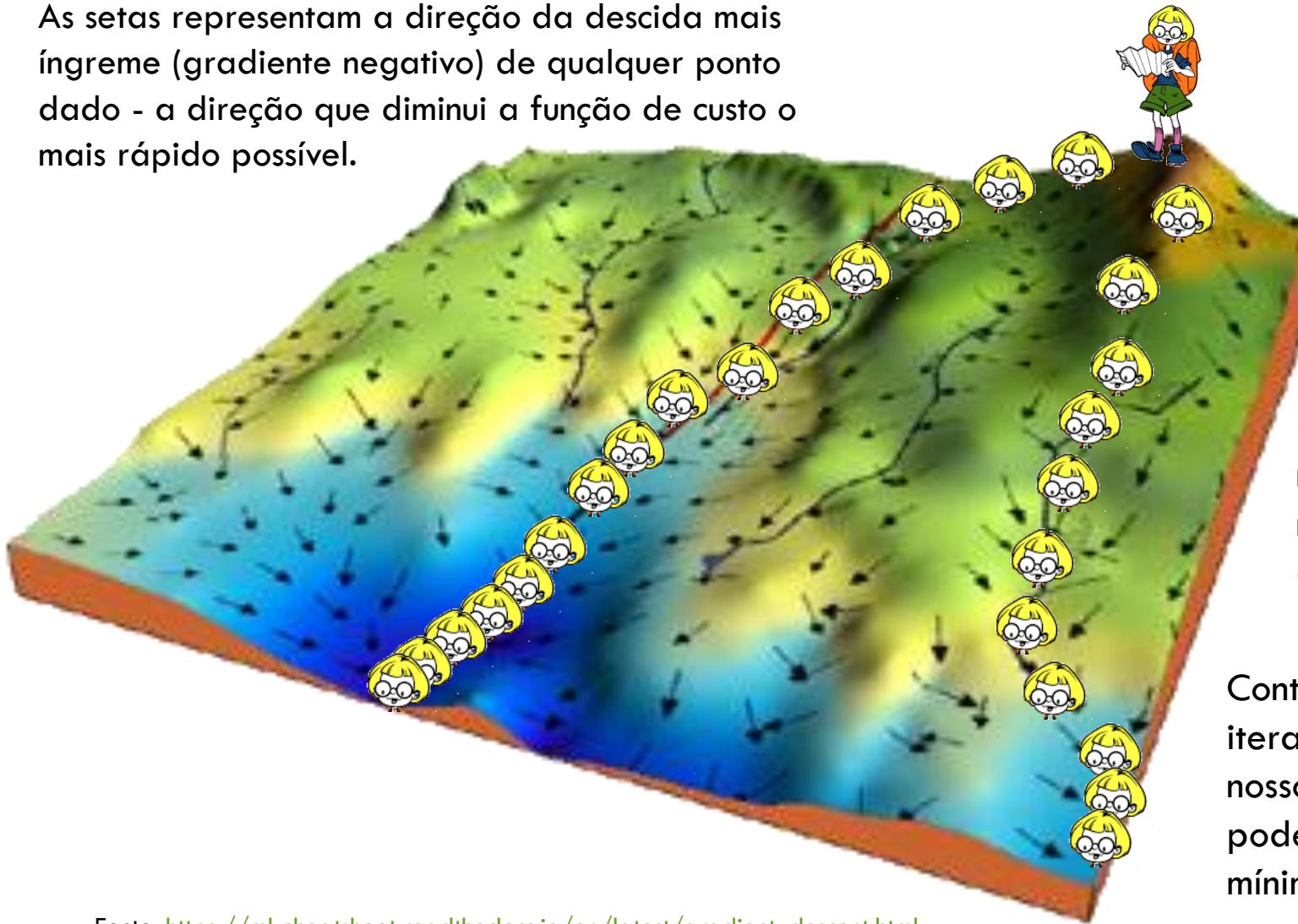
$$\min_{(\omega)} f(\omega)$$

O Gradiente Descendente (GD) é um algoritmo utilizado para encontrar o mínimo de uma função de forma iterativa.



Nosso objetivo é: partindo do alto da montanha, no canto superior direito (alto custo), chegar no mar azul escuro, no canto inferior esquerdo (baixo custo).

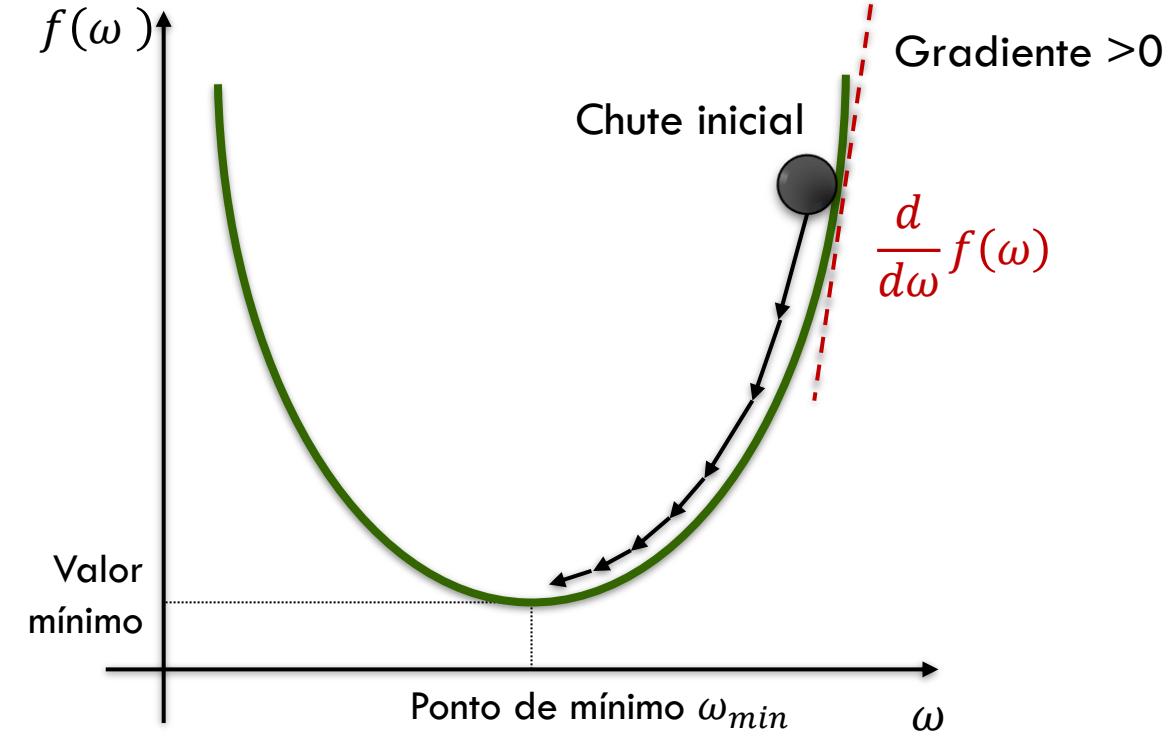
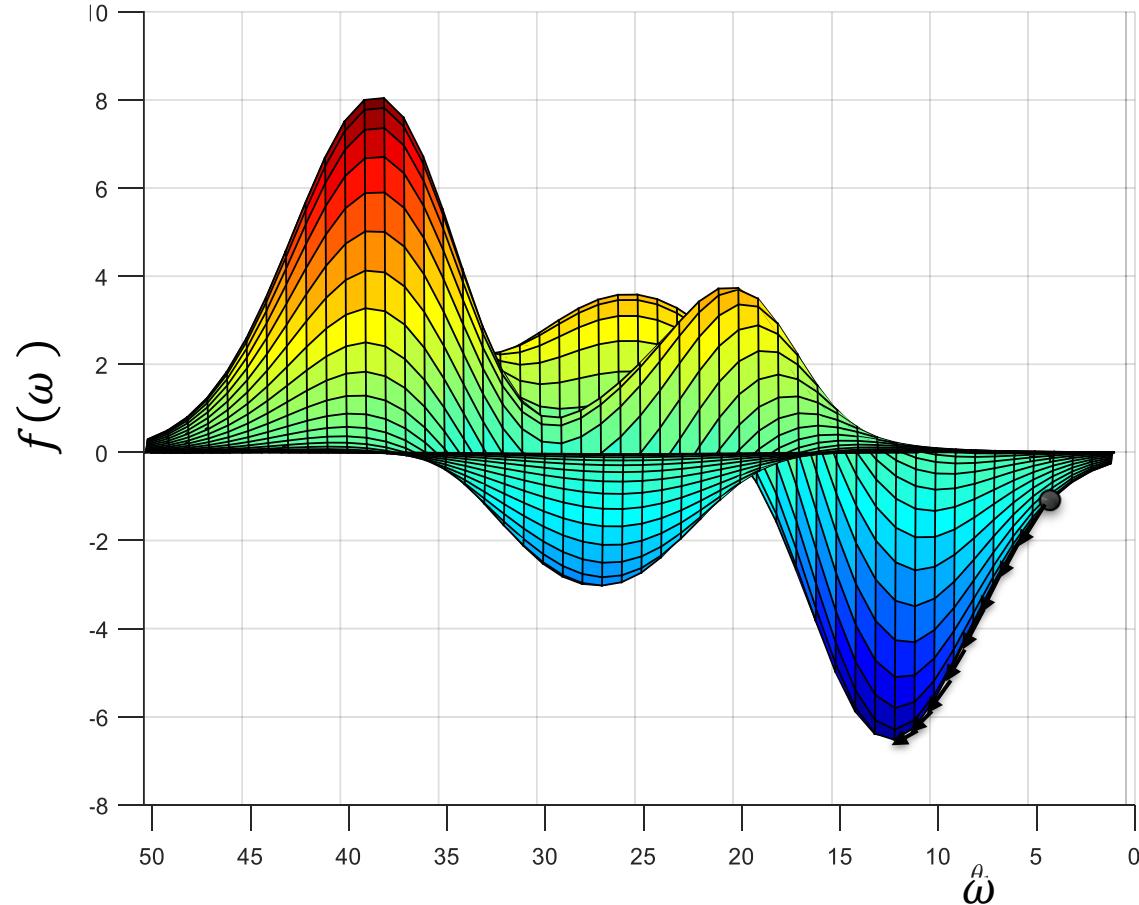
As setas representam a direção da descida mais íngreme (gradiente negativo) de qualquer ponto dado - a direção que diminui a função de custo o mais rápido possível.



Damos nosso primeiro passo em declive na direção especificada pelo gradiente negativo.

Em seguida recalculamos o gradiente negativo (através das coordenadas do nosso novo ponto) e damos outro passo na direção que ele especifica.

Continuamos este processo de forma iterativa até chegarmos ao fundo do nosso gráfico, ou a um ponto em que não podemos mais nos mover para baixo - um mínimo local.



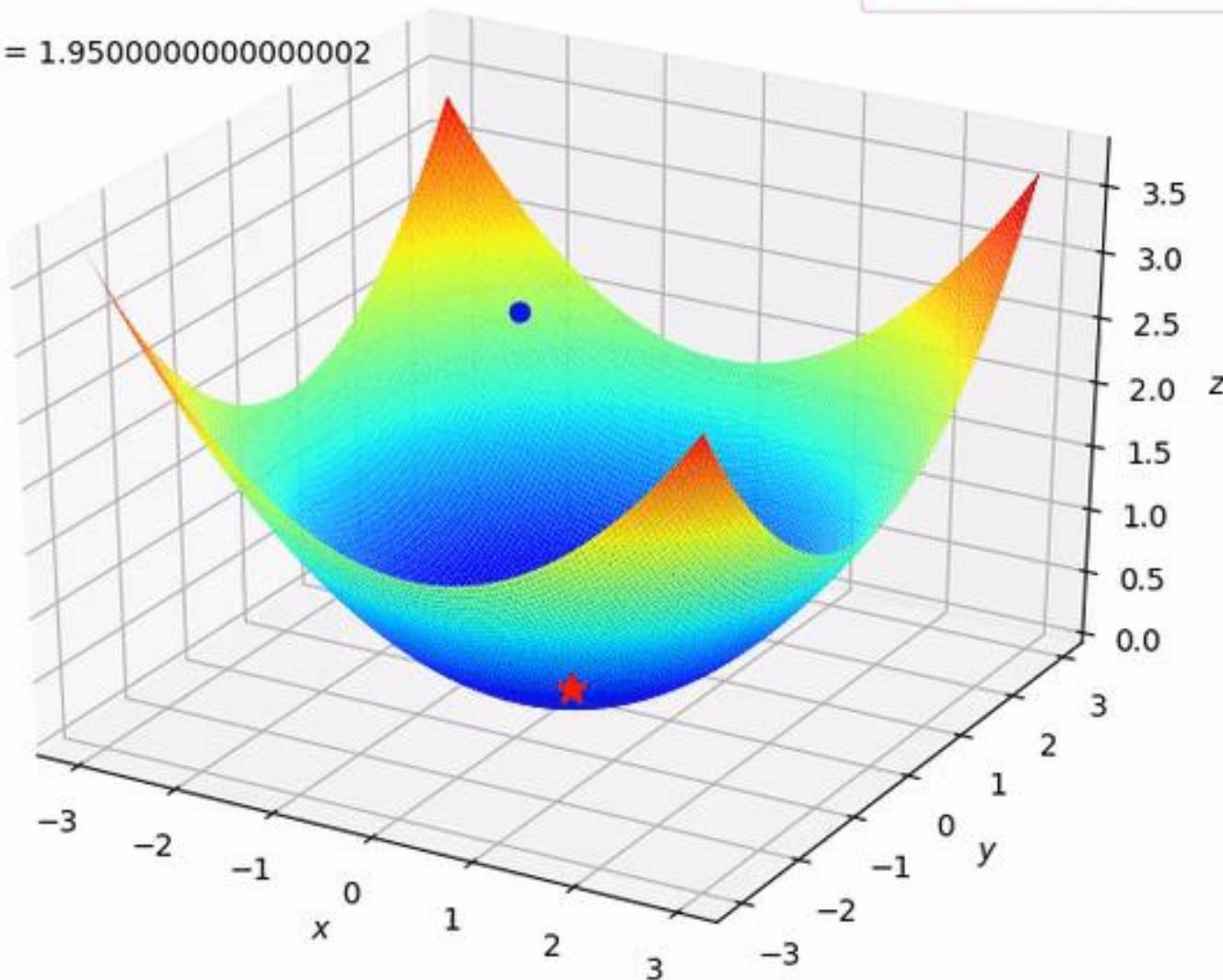
Enquanto $\|\nabla f(\omega)\| > \varepsilon$:

$$\begin{aligned}\omega^{(i+1)} &:= \omega^{(i+1)} - \alpha \nabla f(\omega^{(i)}) \\ i &+= 1\end{aligned}$$

Começamos com um ponto aleatório na função e nos movemos **na direção negativa do gradiente** da função para alcançar os mínimos locais/globais.

Min = 1.9500000000000002

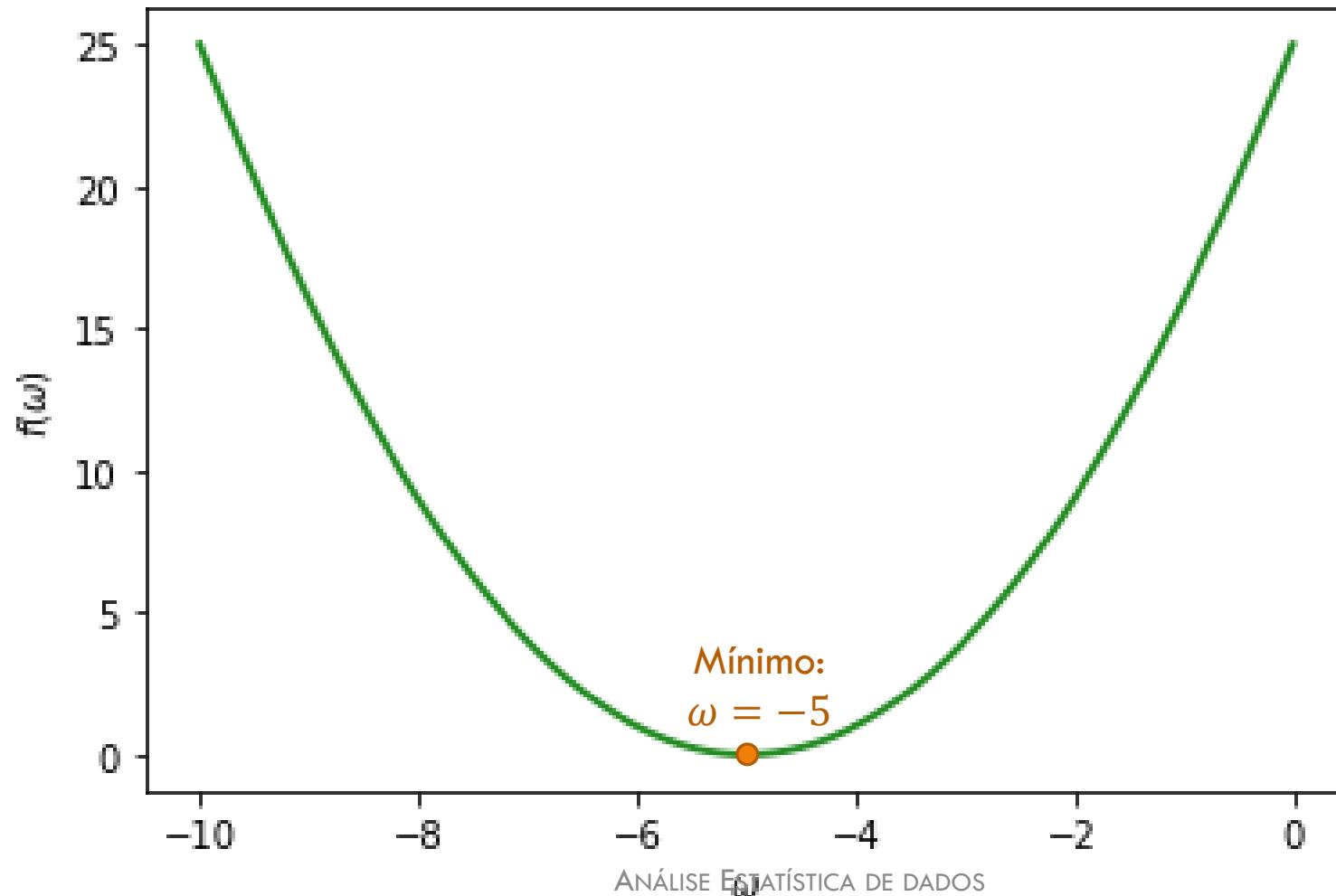
Gradient descent



Fonte: <https://medium.com/@lucasoliveiras/regress%C3%A3o-linear-do-zero-com-python-ef74a81c4b84>

EXEMPLO

$$f(\omega) = (\omega + 5)^2$$



Encontre o valor
do mínimo da
função $\omega = -5$
Usando o
método do
Gradiente
Descendente
que você
aprendeu.

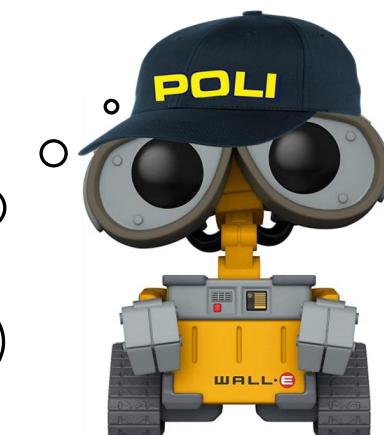
ALGORITMO DE GRADIENTE DESCENDENTE

1. Inicialização dos parâmetros: Atribua um valor inicial ω_0 para os parâmetros;
2. Cálculo do Jacobiano da função;
3. Atualização dos parâmetros na direção oposta ao gradiente de forma a minimizar o valor da função

$$\omega^{(j+1)} = \omega^{(j)} - \alpha \frac{\partial J(\omega)}{\partial \omega^{(j)}}$$

Repetição das etapas 2 e 3 até a obtenção de um valor desejado.

O que é
um valor
desejado?



QUANDO PARAR?

Quando a variação $\omega^{(j+1)} = \omega^{(j)}$ for menor que uma precisão ε . Por exemplo, $\varepsilon = 0,000001$.

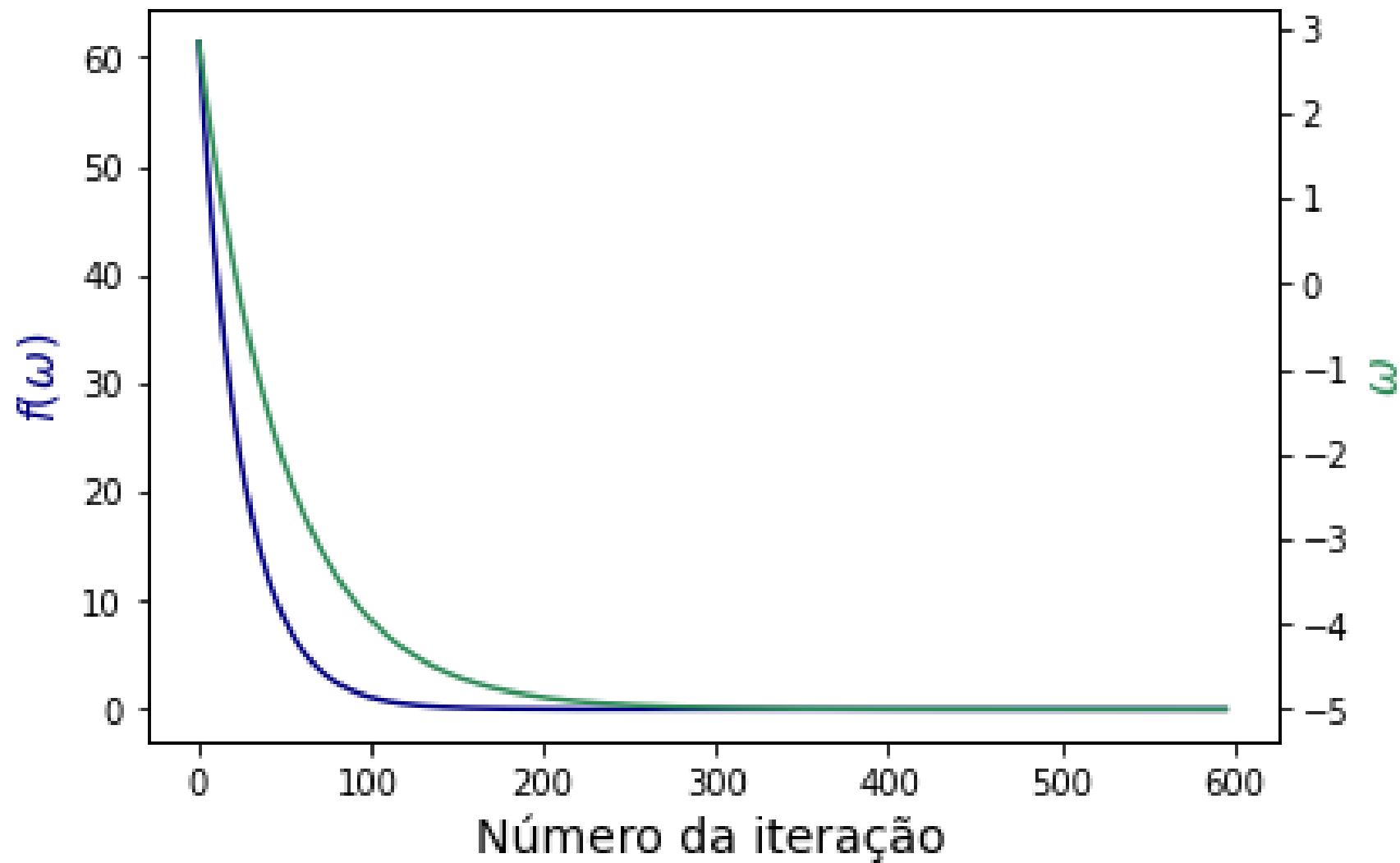
Quando o número de iterações ultrapassa um limite T . Por exemplo, $T = 10000$.

VAMOS SEGUIR OS PASSOS:

$$f(\omega) = (\omega + 5)^2, \quad \frac{df}{d\omega} = 2(\omega + 5)$$

1. Inicializamos com $x = 3$.
2. Modificação de ω na direção do negativo do gradiente. Quanto mover? Para isso, precisamos da taxa de aprendizado. Vamos supor que $\alpha \leftarrow 0,01$,

Lentamente
a caminho
do mínimo...



TAXA DE APRENDIZAGEM α

Taxa de aprendizagem α controla o tamanho do passo em cada iteração.

Selecionar o valor correto é uma decisão crítica do modelo:

- Pode convergir para um mínimo local se a função de custo não for convexa;
- Quando nos aproximamos de um mínimo local, a descida de gradiente tomará automaticamente passos menores. Portanto, não há necessidade de diminuir α ao longo do tempo;
- Valores de α não podem ser muito grandes, nem muito pequenos...

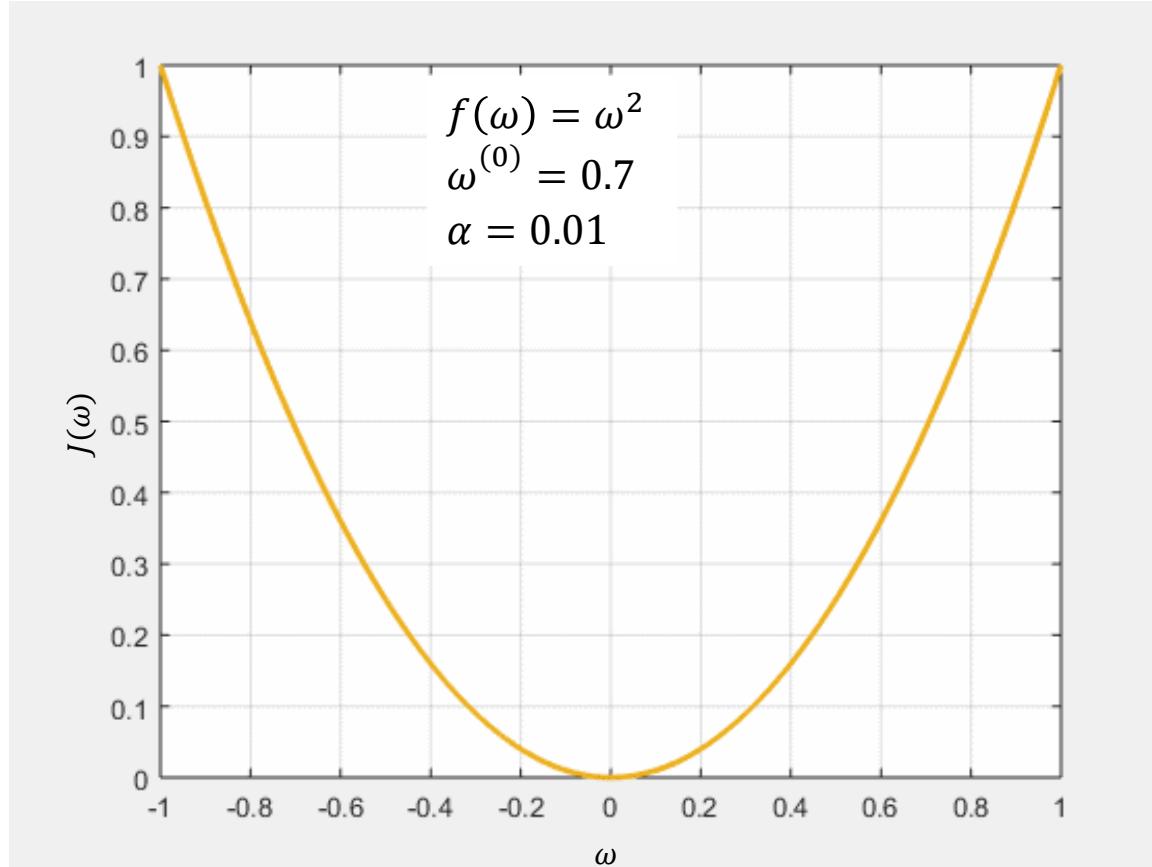
Tente ... 0,001 – 0,003 – 0,01 – 0,03 – 0,1 – 0,3 – 1 ...

Suba/Desça ~ 3x em 3x

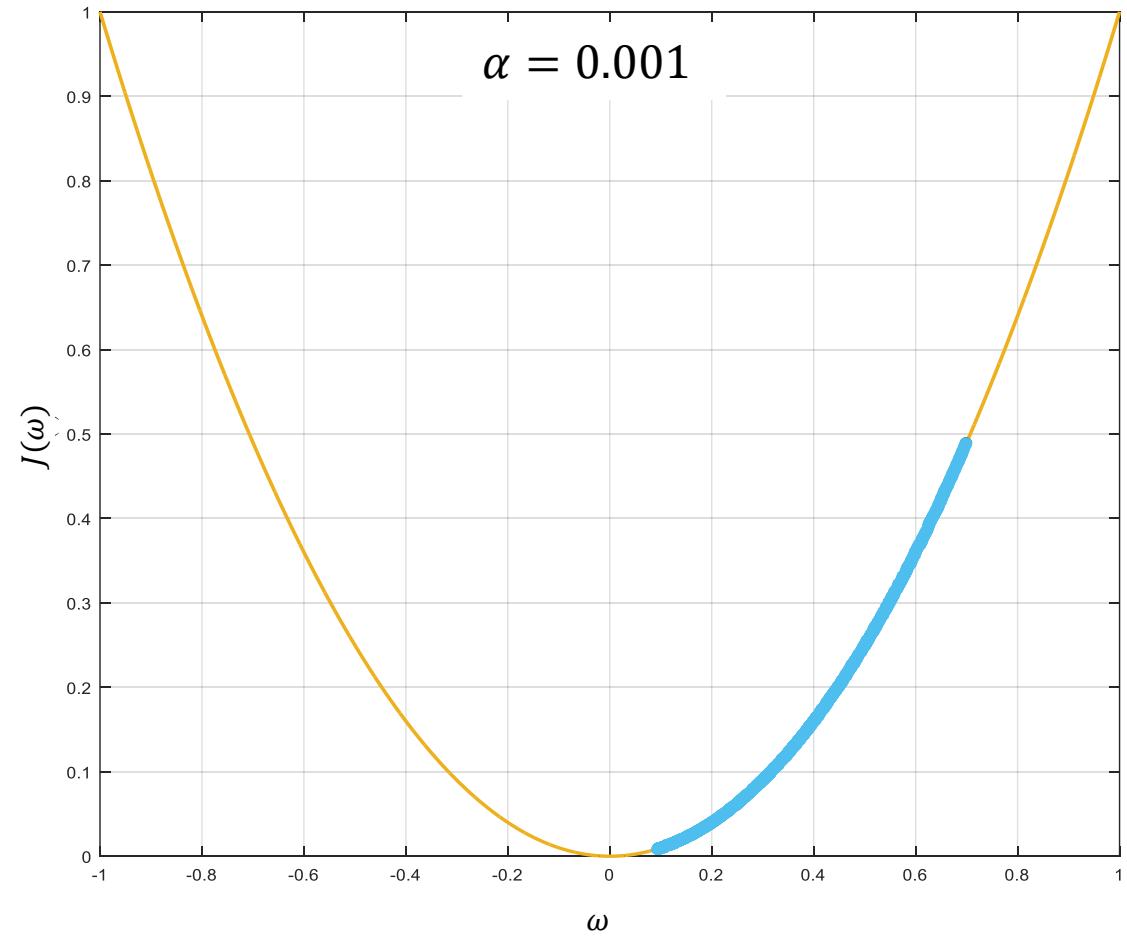
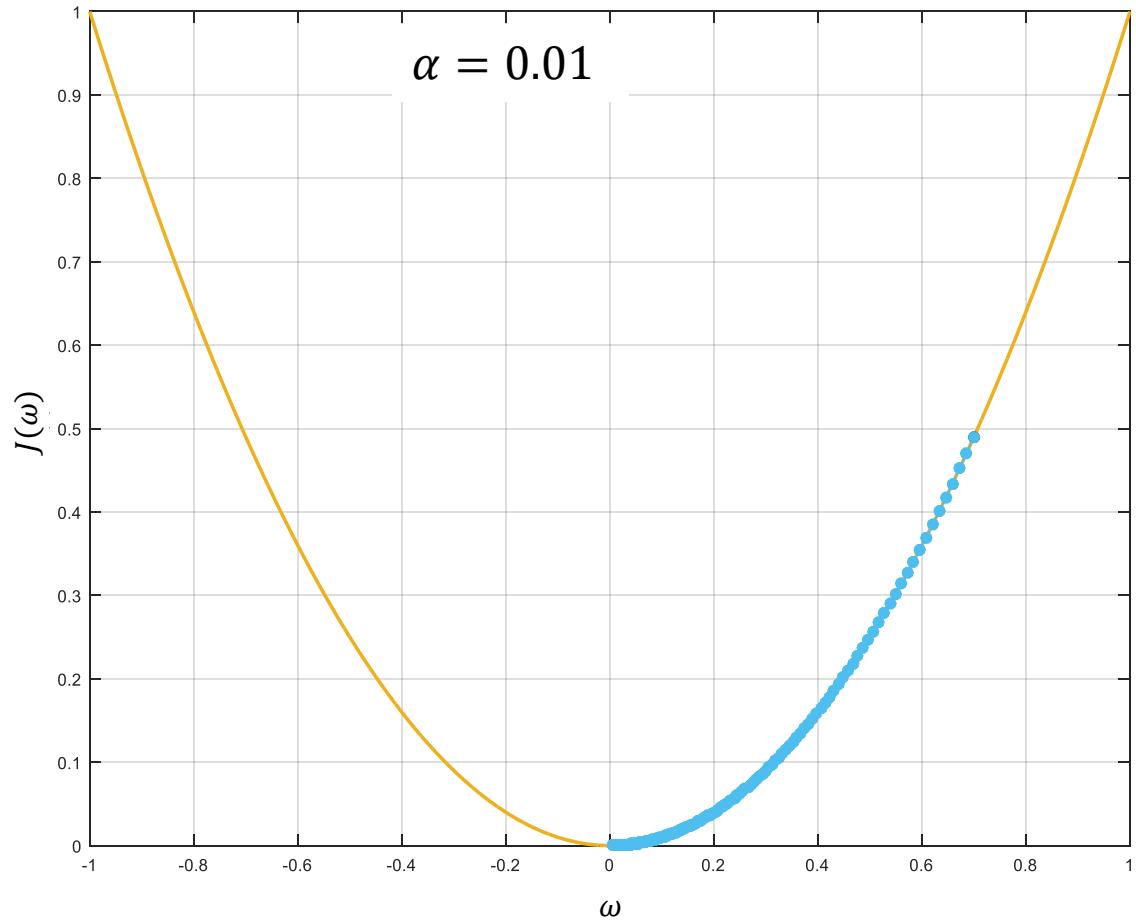


α MUITO PEQUENO

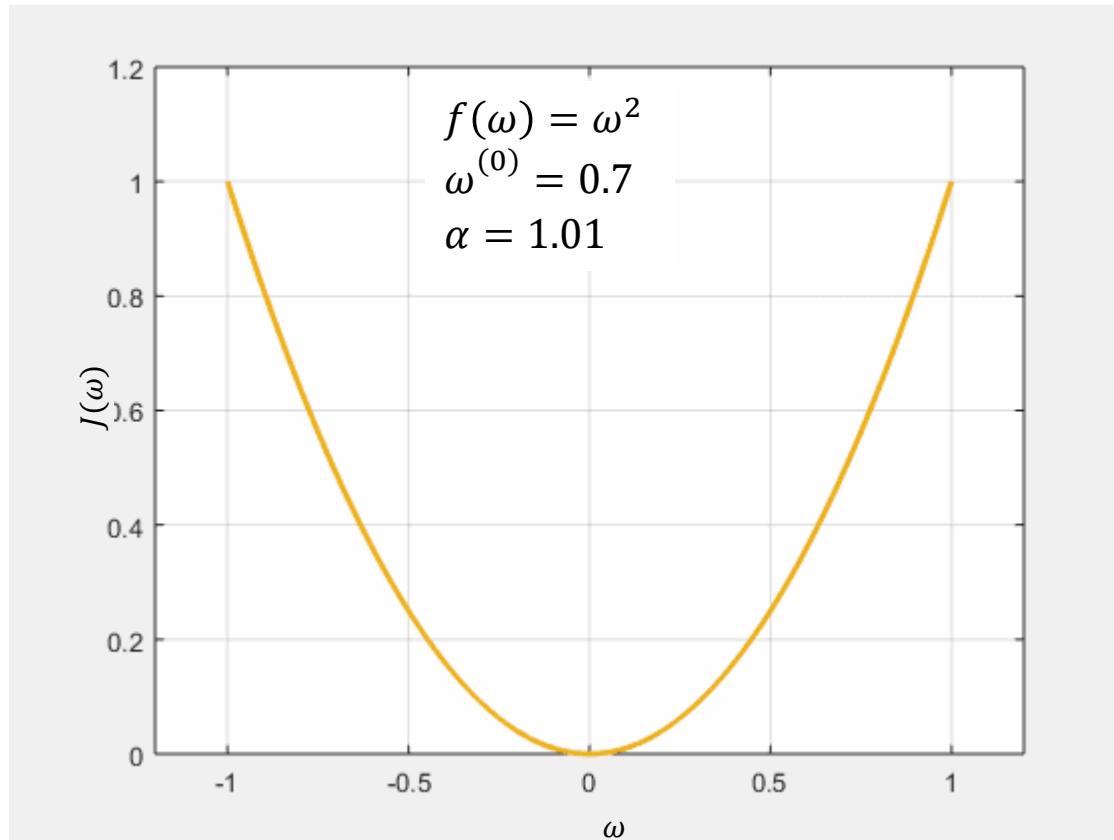
Vamos simplificar a função de custo como algo da forma $f(\omega) = \omega^2$



Uma taxa de aprendizado muito baixa é mais precisa, mas o cálculo do gradiente é demorado, então levará muito tempo para chegarmos ao fim. ... Podemos nos mover com confiança na direção do gradiente negativo, pois estamos recalculando-o com muita frequência, mas “desceremos muito devagar”;



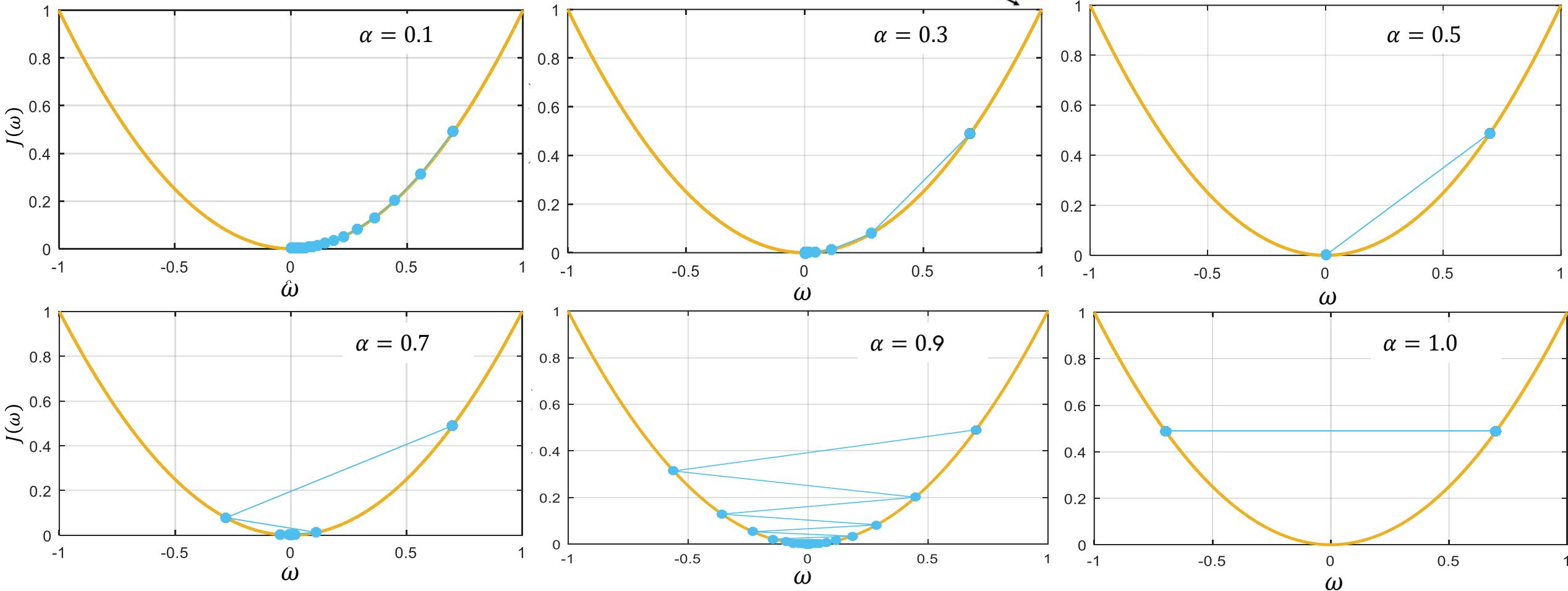
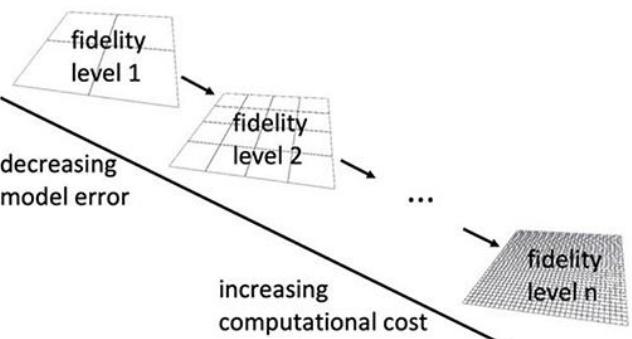
α MUITO GRANDE



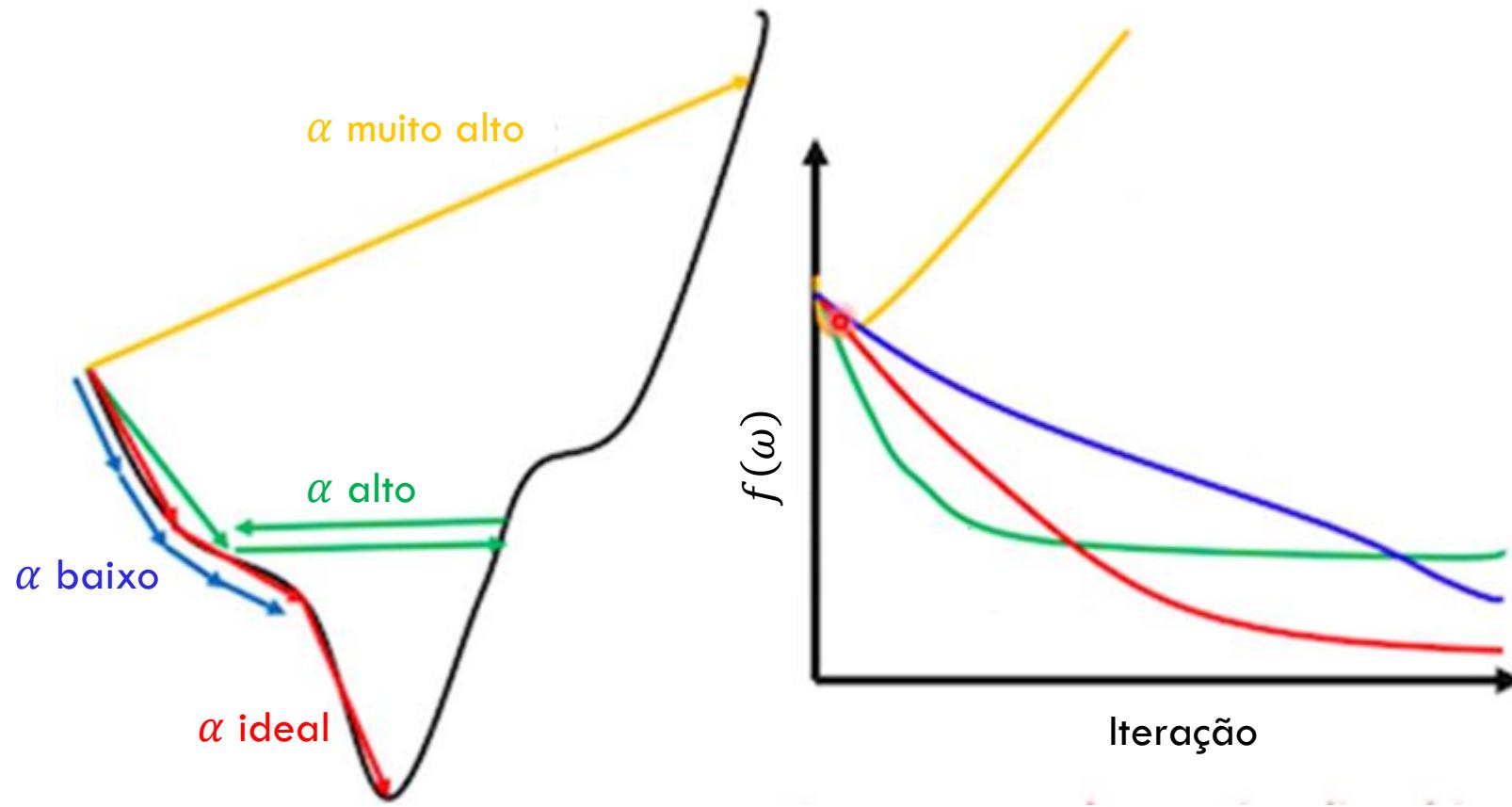
Se utilizarmos uma taxa de aprendizado muito alta, podemos cobrir maior distância a cada passo, mas corremos o risco de ultrapassar o ponto mais baixo, já que o gradiente está mudando constantemente. O método diverge porque estamos “descendo muito rápido”.

$$f(\omega) = \omega^2$$

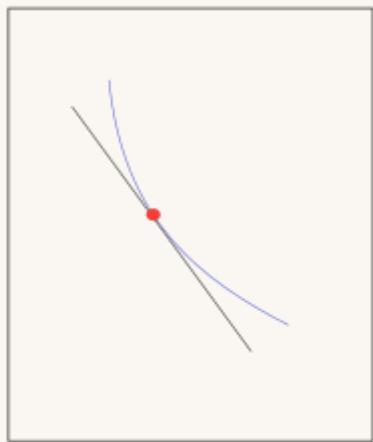
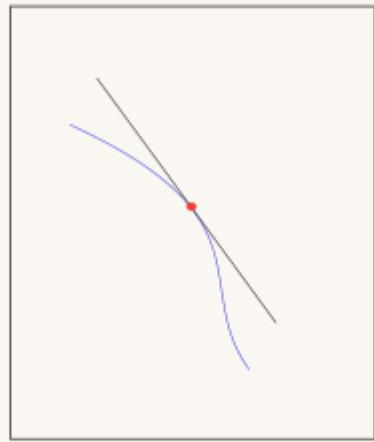
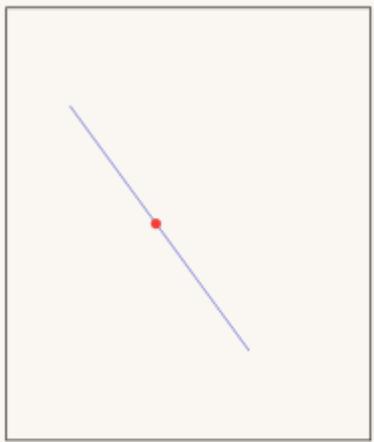
$$\omega^{(0)} = 0.7$$



α IDEAL



Diminui acentuadamente e
depois se torna mais suave...

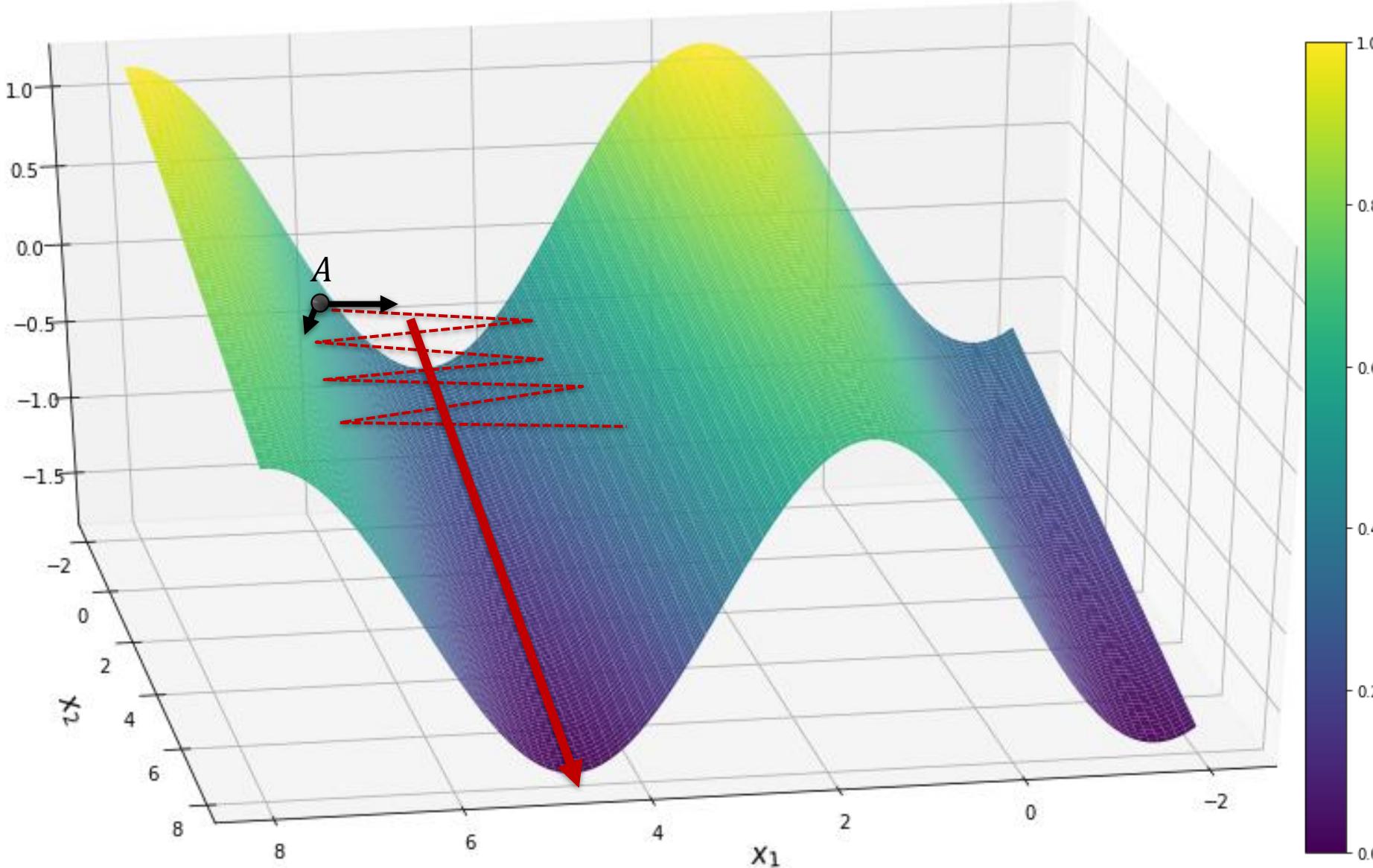


"All of the these curves are the same."

"What ar... wait...is that you Gradient Descent?"

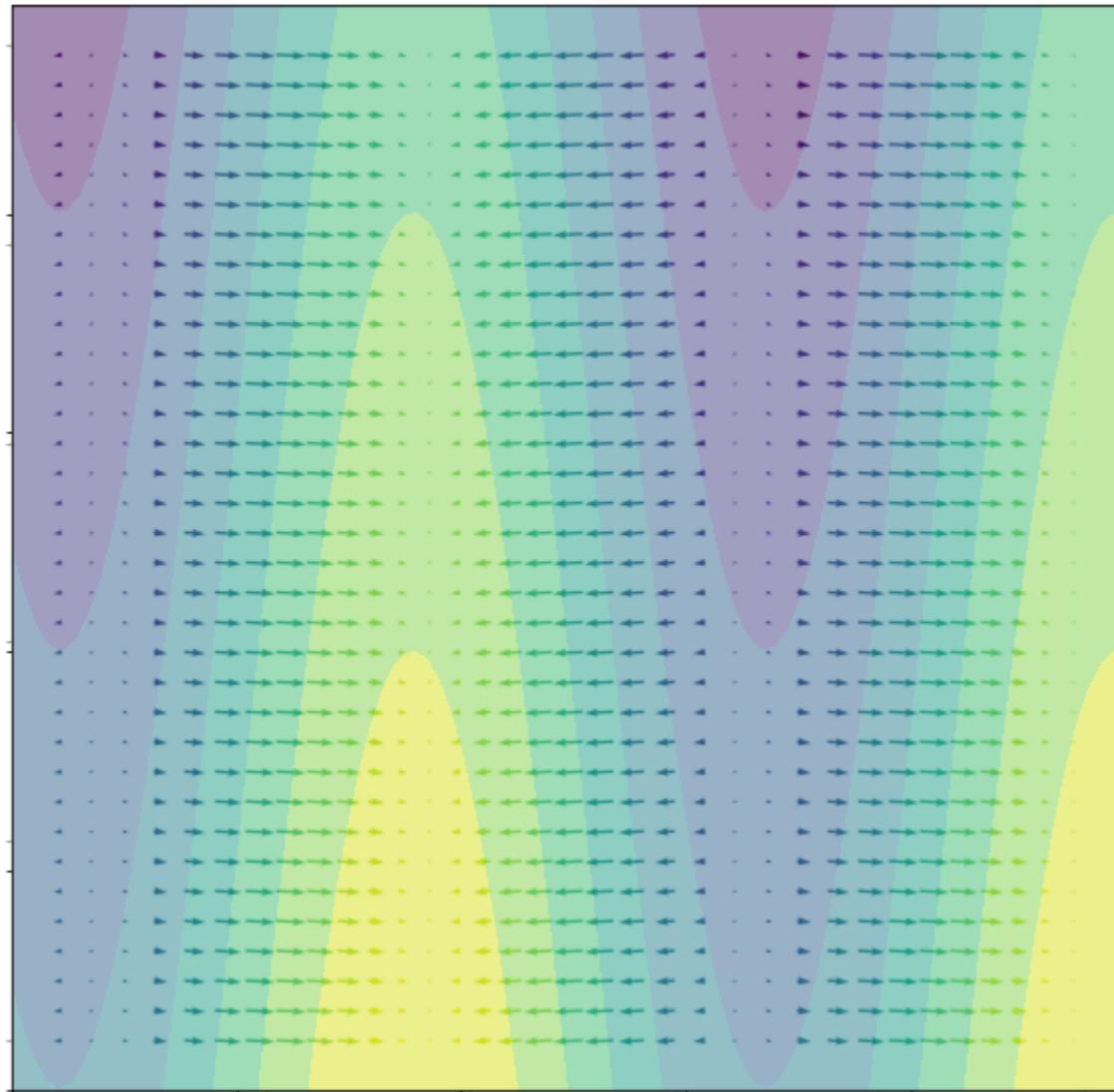
"Gradient descent is a First Order Optimization Method. It only takes the first order derivatives of the loss function into account and not the higher ones. What this basically means it has no clue about the curvature of the loss function. **It can tell whether the loss is declining and how fast, but cannot differentiate between whether the curve is a plane, curving upwards or curving downwards."**

<https://blog.paperspace.com/intro-to-optimization-momentum-rmsprop-adam/>



Um problema com o algoritmo de gradiente descendente é que a busca na direção mais inclinada, ou seja, na direção perpendicular às curvas de nível pode levar a um zig-zag e a convergência será muito lenta.

Por exemplo, se começarmos a partir do ponto A, a direção perpendicular à curva de nível aponta numa direção quase 90° graus da direção ao ponto de mínimo.



SOLUÇÃO É A HESSIANA?

Já sabemos que a **hessiana** é uma **matriz** que organiza todas as derivadas parciais de segunda ordem de uma função.

PORÉM, requer que você calcule gradientes da função de perda em relação a cada combinação de parâmetros ω_i, ω_j . Para problemas modernos, o número de parâmetros pode estar em bilhões, e ter que calcular um bilhão de gradientes quadrados torna computacionalmente intratável o uso de métodos de otimização de ordem superior.

NO ENTANTO, a otimização de segunda ordem consiste em incorporar as informações sobre **como o gradiente está mudando**. Embora não possamos calcular com precisão essas informações, **podemos escolher seguir heurísticas que orientam nossa busca por ótimos com base no comportamento passado do gradiente**.

MOMENTUM

O Momentum propõe o seguinte ajuste para a descida gradiente.

$$m = \beta m - \alpha J(\omega)$$

$$\omega = \omega + m$$

m é o gradiente que é mantido nas iterações anteriores. Este gradiente retido é multiplicado por um valor denominado "Coeficiente de Momentum" β , que é a porcentagem do gradiente retido a cada iteração.

Se definirmos o valor inicial de m como 0 e escolhermos nosso coeficiente como 0.9, as equações de atualização subsequentes serão,

$$m_1 = -G^{(1)}$$

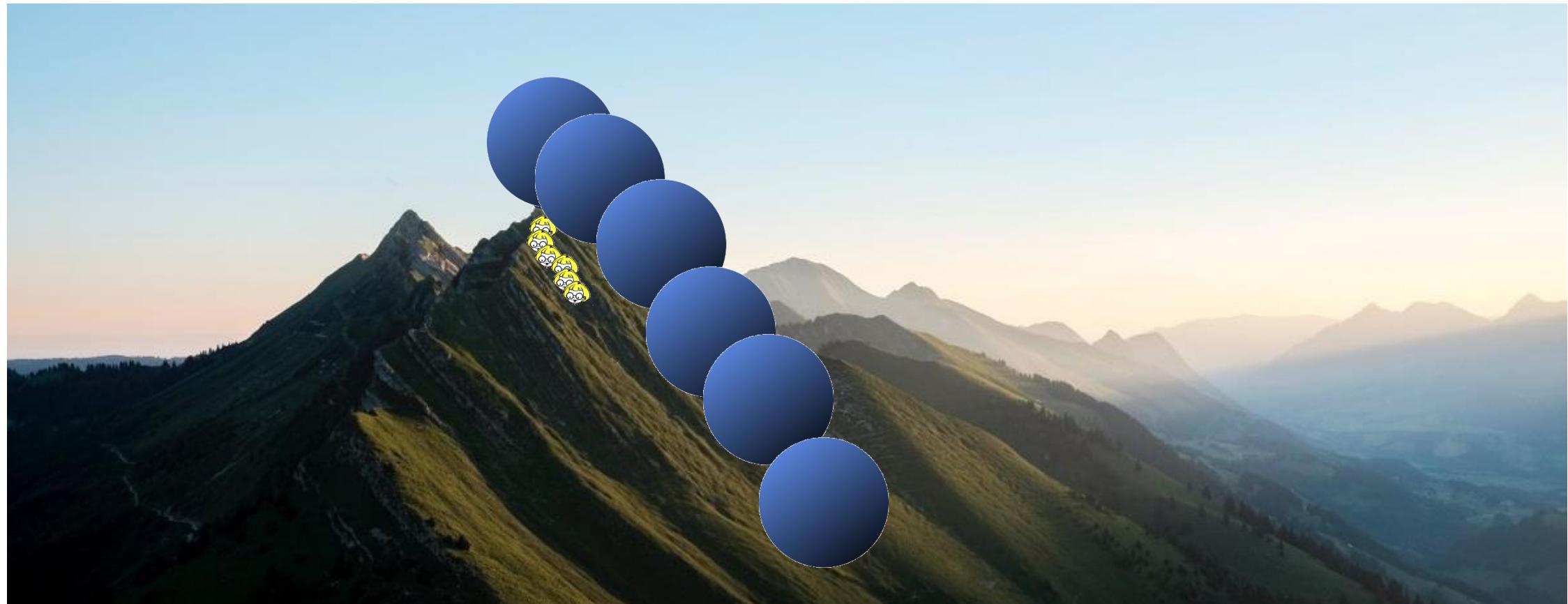
$$m_2 = -0.9G^{(1)} - G^{(2)}$$

$$G^{(i)} = \alpha J(\omega^{(i)})$$

$$m_3 = -0.9(0.9G^{(1)} + G^{(2)}) - G_3 = -0.81G^{(1)} - 0.9G^{(2)} - G^{(3)}$$

Damos ao gradiente descendente uma memória de curto prazo!

GD vs GD COM MOMENTO



Gradient descent with momentum

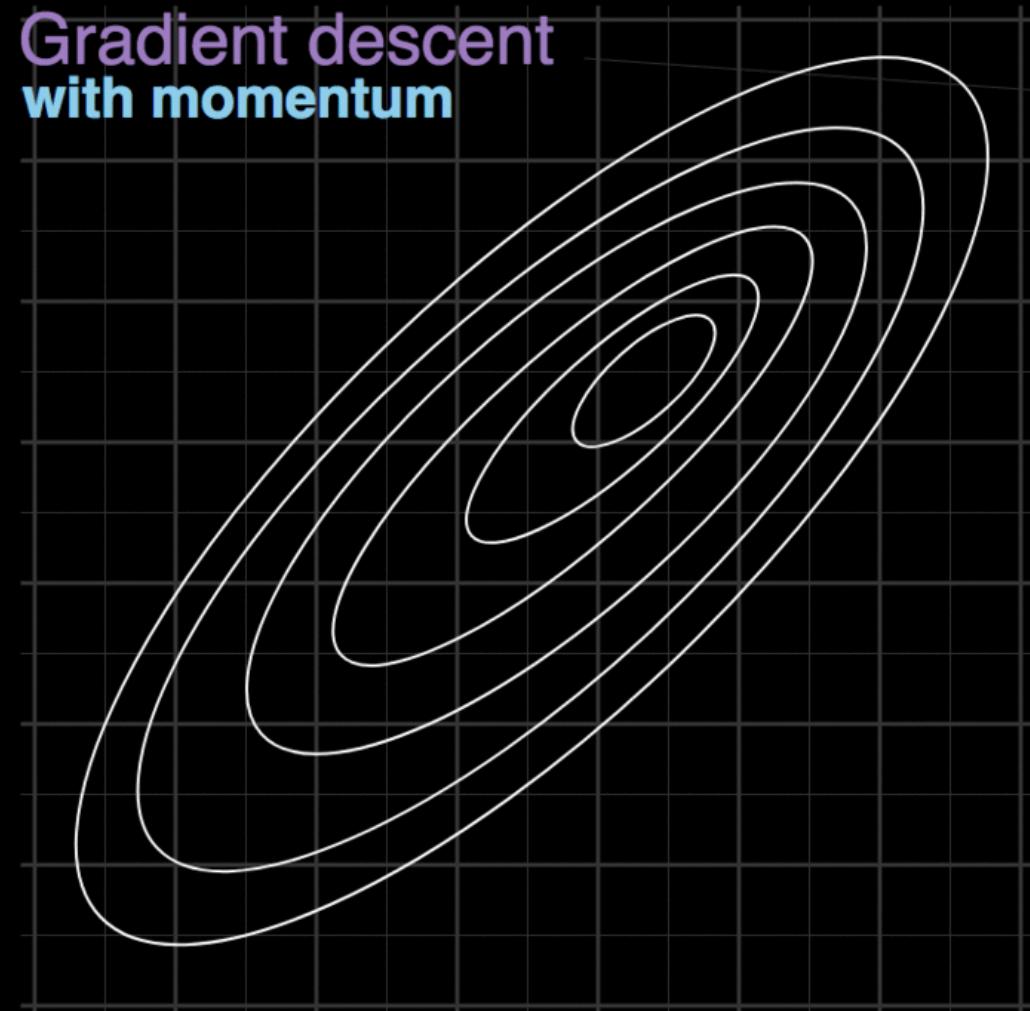


Figura extraída de:
<https://www.itread01.com/content/1543467366.html>

LIÇÃO DE CASA

Ache o vetor ω para a regressão linear do conjunto de dados da tabela ao lado. Na equação,

$$y = \omega_0 x_0 + \omega_1 x_1 + \omega_2 x_2$$

y é a taxa de gordura no sangue, ω são os pesos e x os dados de entrada.

Fonte:

<https://people.sc.fsu.edu/~jburkardt/datasets/regression/x09.txt>

Índice	Peso (Kg)	Idade (anos)	Taxa de gordura no sangue
1	84	46	354
2	73	20	190
3	65	52	405
4	70	30	263
5	76	57	451
6	69	25	302
7	63	28	288
8	72	36	385
9	79	57	402
10	75	44	365
11	27	24	209
12	89	31	290
13	65	52	346
14	57	23	254
15	59	60	395
16	69	48	434
17	60	34	220
18	79	51	374
19	75	50	308
20	82	34	220
21	59	46	311
22	67	23	181
23	85	37	274
24	55	40	303
25	63	30	244



ACABOU...

Revejam o material
disponibilizado em aula.
Refaçam exercícios.
Até a próxima semana!!!