ICC204 - Aprendizagem de Máquina e Mineração de Dados

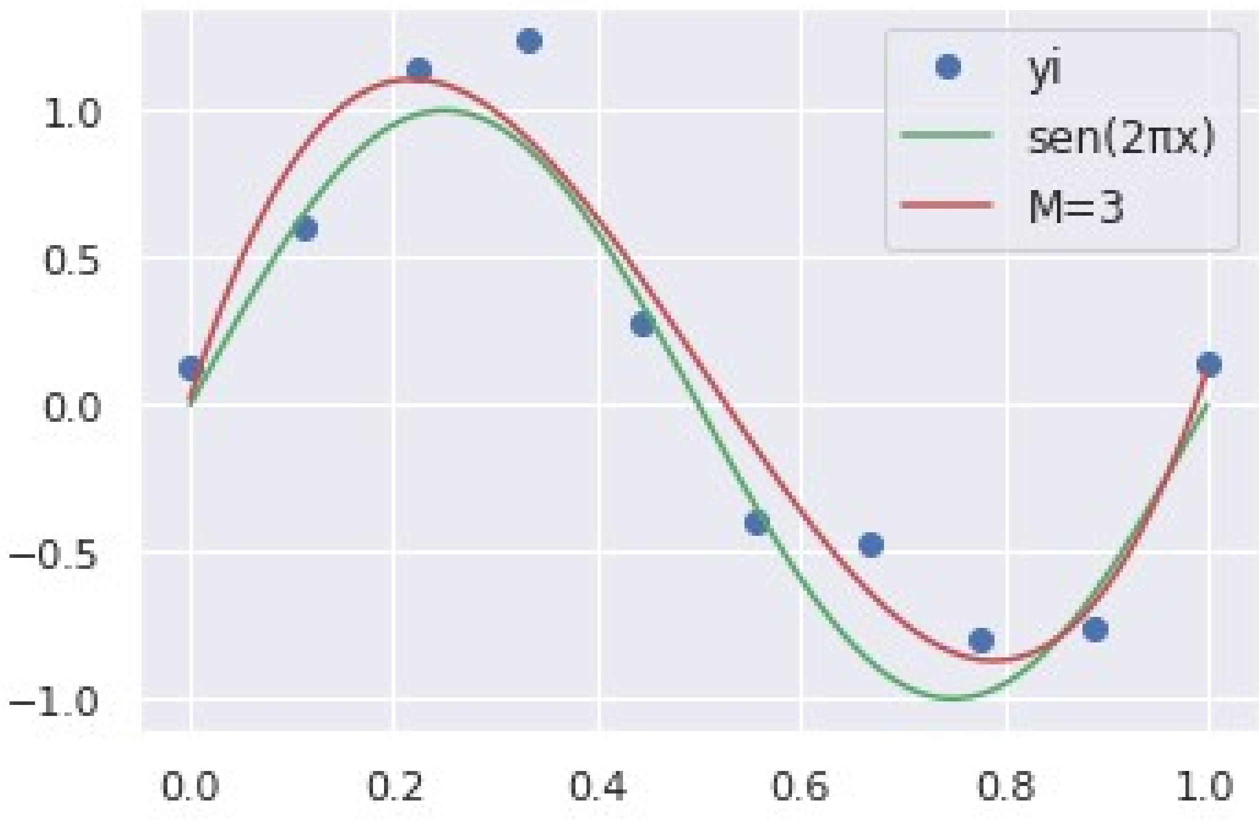
Regressão e Método do Gradiente

Instituto de Computação

Prof. Rafael Giusti
rgiusti@icomp.ufam.edu.br

Regressão

- Como vimos no começo do semestre, modelos de **regressão** são modelos de aprendizado supervisionado nos quais o rótulo é um atributo numérico



Exemplos de treinamento

Distribuição subjacente dos dados

Modelo regressor

Regressão

- Se assumirmos que o nosso modelo de regressão é linear, então ele terá a seguinte forma

$$y(x, \mathbf{w}) = \mathbf{w} \cdot x$$

- Expandindo o produto escalar $\mathbf{w} \cdot x$, temos

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M$$

Regressão

- Se assumirmos que o nosso modelo de regressão é linear, então ele terá a seguinte forma

$$y(x, \mathbf{w}) = \mathbf{w} \cdot x$$

- Podemos encontrar w representando um conjunto de dados como uma matriz $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$, de modo que o conjunto ótimo de pesos $\mathbf{w}^*$ é dado por

$$\mathbf{w}^* = X^{-1}\mathbf{y}$$

Regressão

- Ou podemos representar a **função de perda**

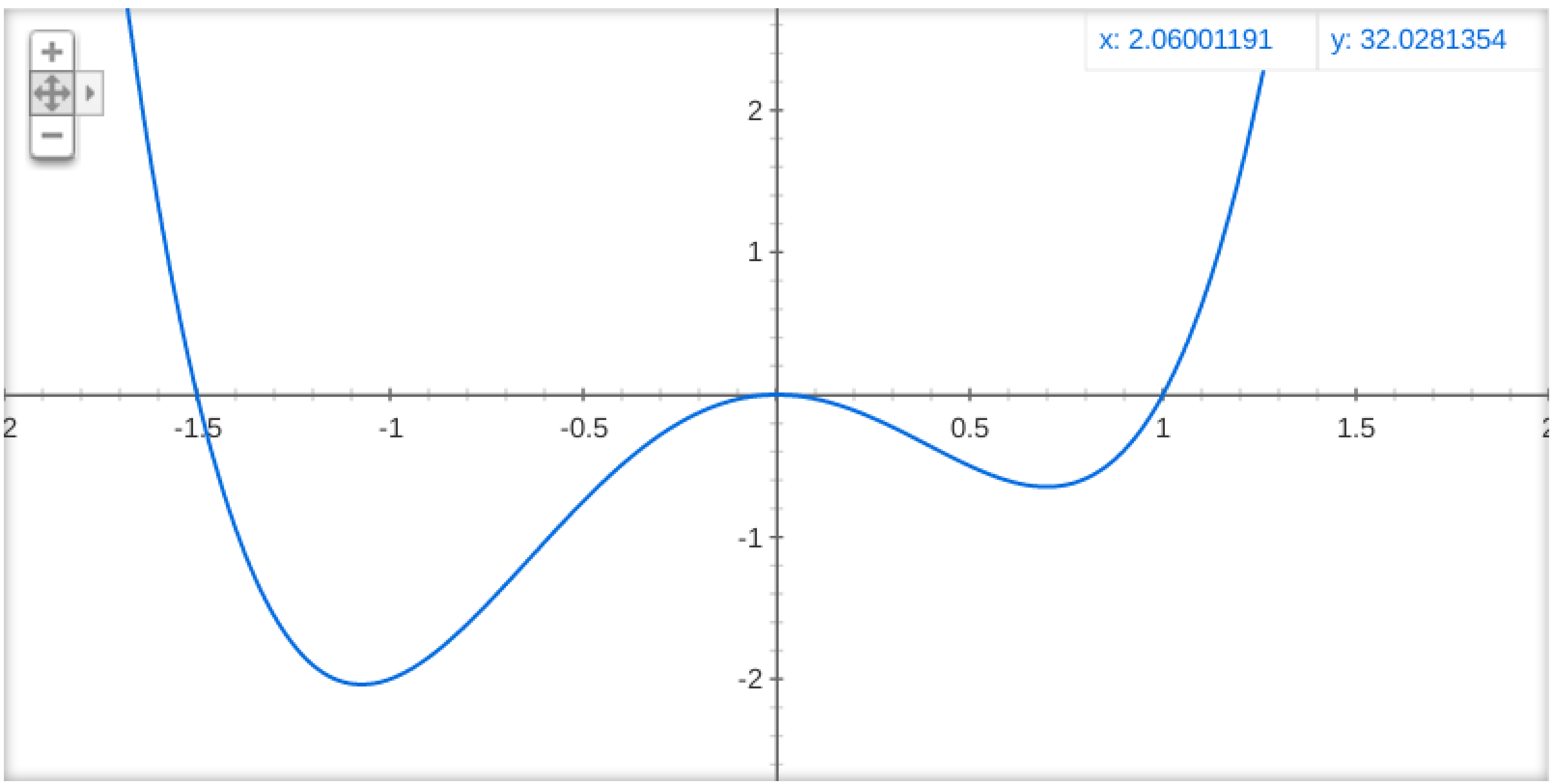
$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}[y(x_i, \mathbf{w}) - t_i]^2$$

e minimizá-la de forma iterativa através (por exemplo) do **método do gradiente**

Minimizando f(x) – uma variável real

- O método do gradiente é uma forma de **minimizar** ou **maximizar** uma função

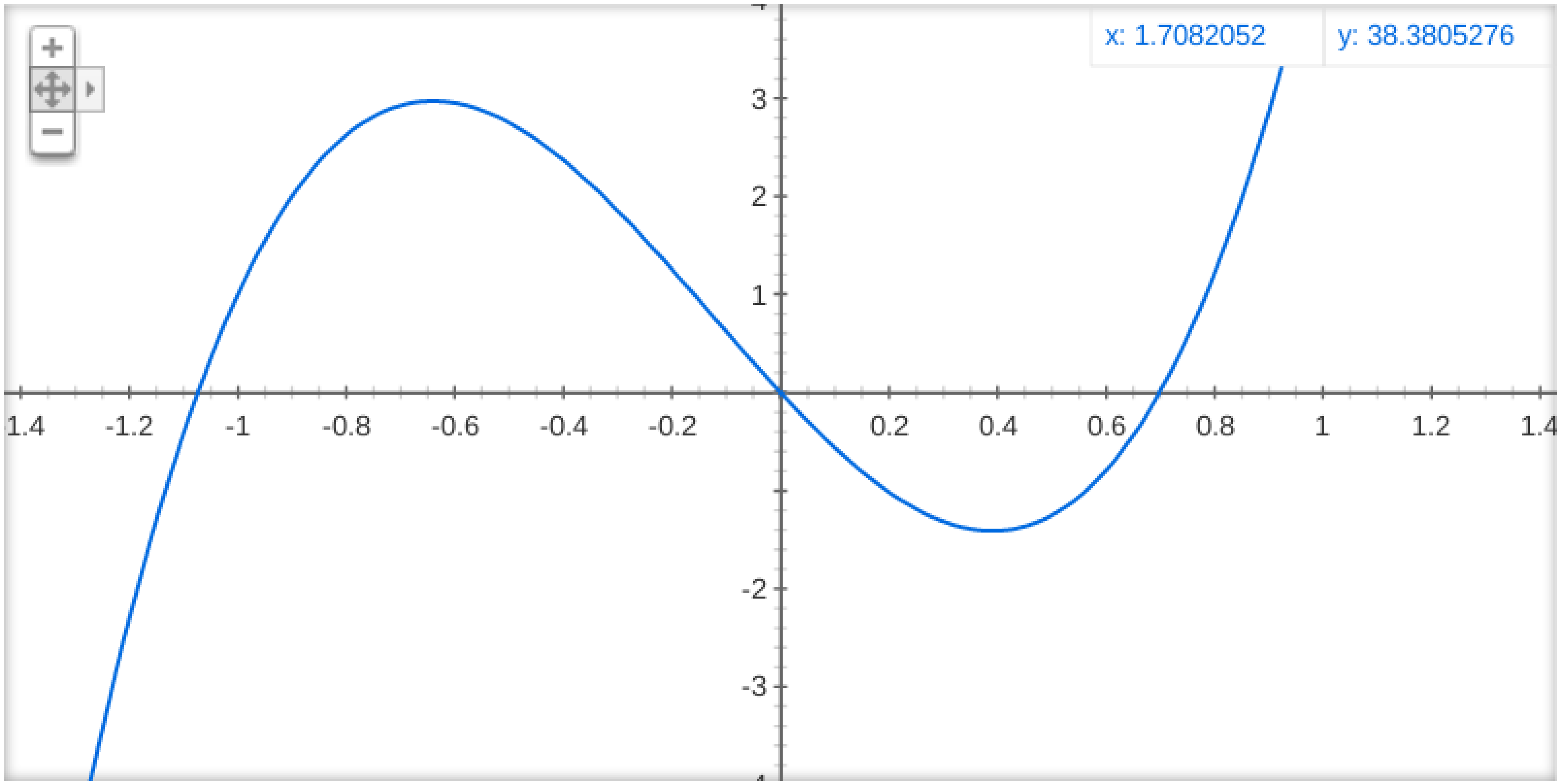
$$f(x) = 2x^4 + x^3 - 3x^2$$



Minimizando f(x) – uma variável real

- Podemos encontrar o mínimo/máximo desse tipo de função verificando a primeira derivada

$$f'(x) = 8x^3 + 3x^2 - 6x$$

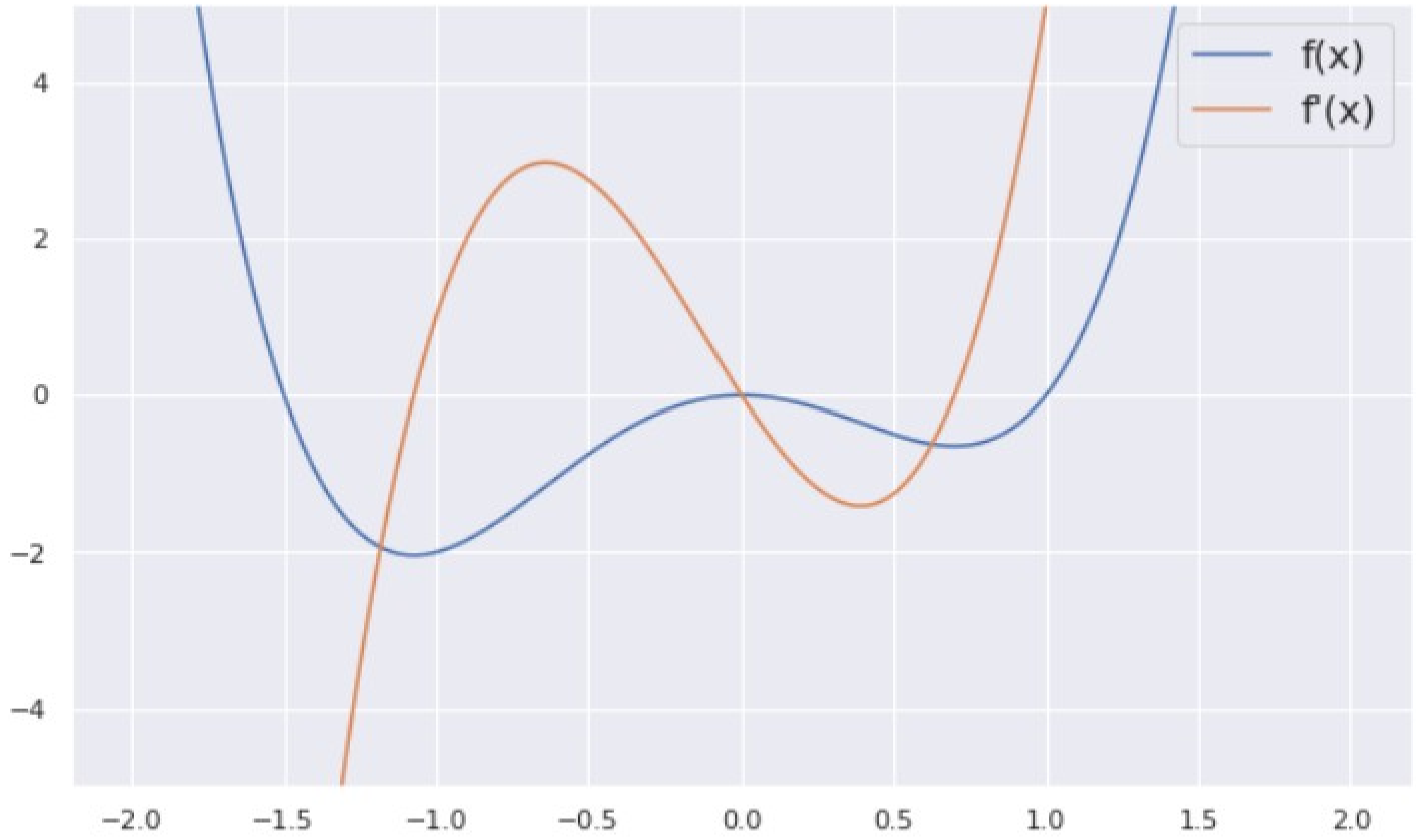


Minimizando f(x) – uma variável real

- Os pontos nos quais $f'(x) = 0$ são os **pontos críticos** da função $f(x)$
- Podem ser
  - Pontos de inflexão
  - Mínimos ou máximos locais
  - Mínimos ou máximos globais
- Mas e se encontrar as raízes de $f'(x)$ não for uma opção?

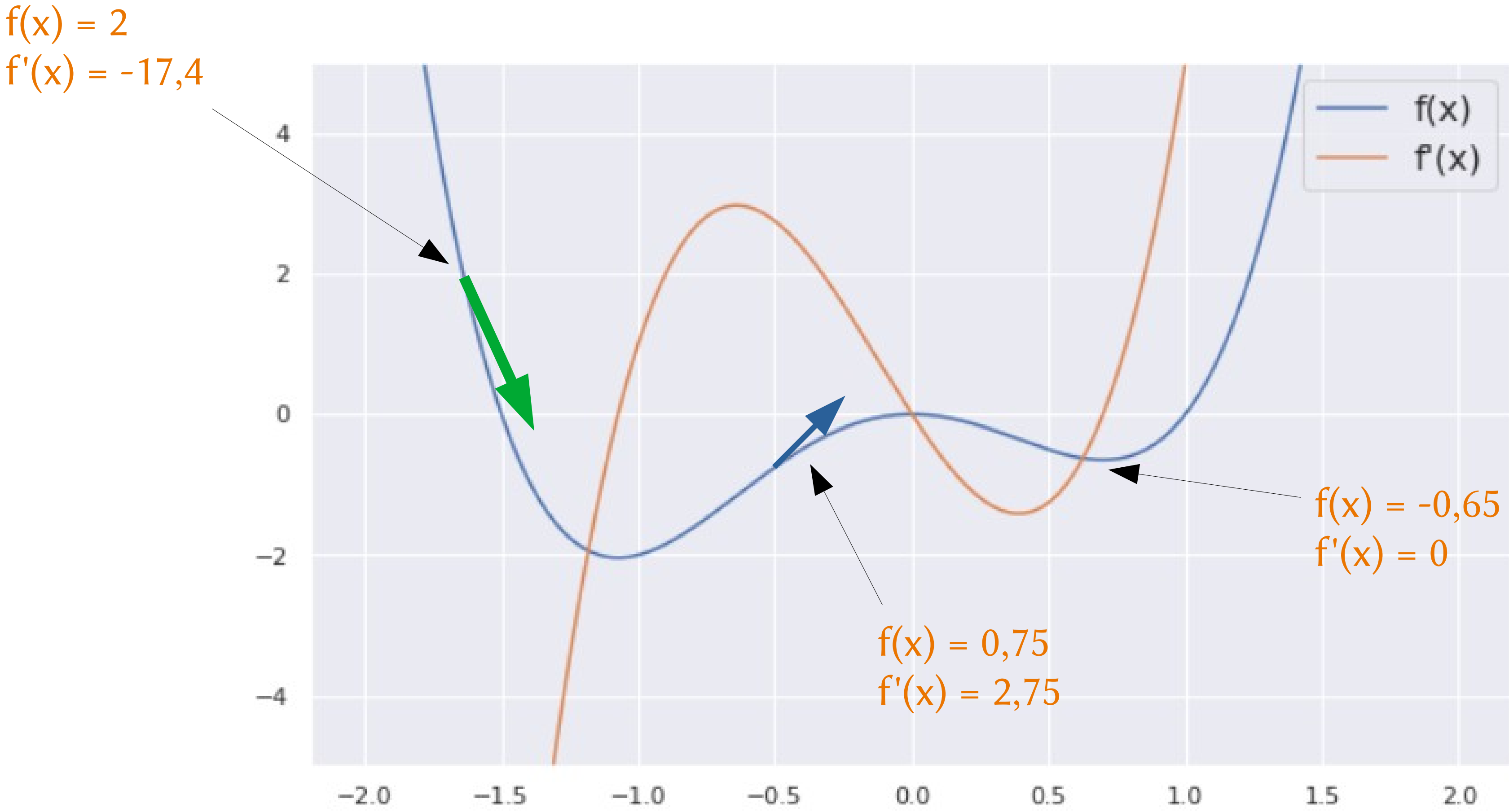
"Descendo" a derivada

- A derivada da função "aponta" para o sentido de "crescimento da função"



"Descendo" a derivada

- A derivada da função "aponta" para o sentido de "crescimento da função"



"Descendo" a derivada

- Conhecendo **f(x)** e **f'(x)** para um valor específico de **x**, podemos nos aproximar de um máximo ou mínimo deslocando no sentido **oposto** da derivada
- Exemplo no *notebook*

Método do gradiente

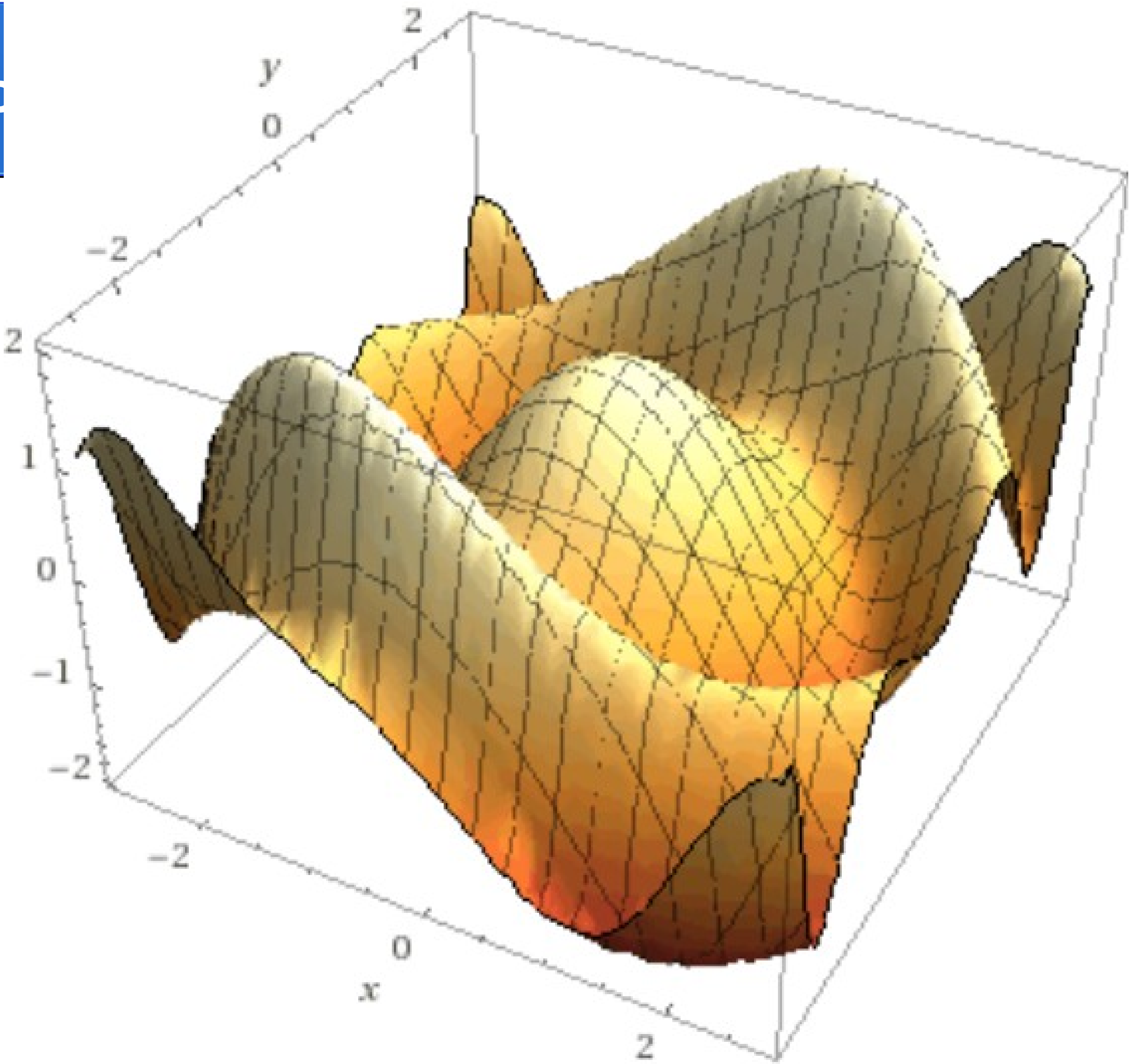
- O gradiente é o conceito generalizado da derivada para uma função que possui múltiplas variáveis

$$f(x,y) = \sin(\frac{1}{2}x^2 + y^2 + 3) + \cos(x - \frac{1}{2}y)$$

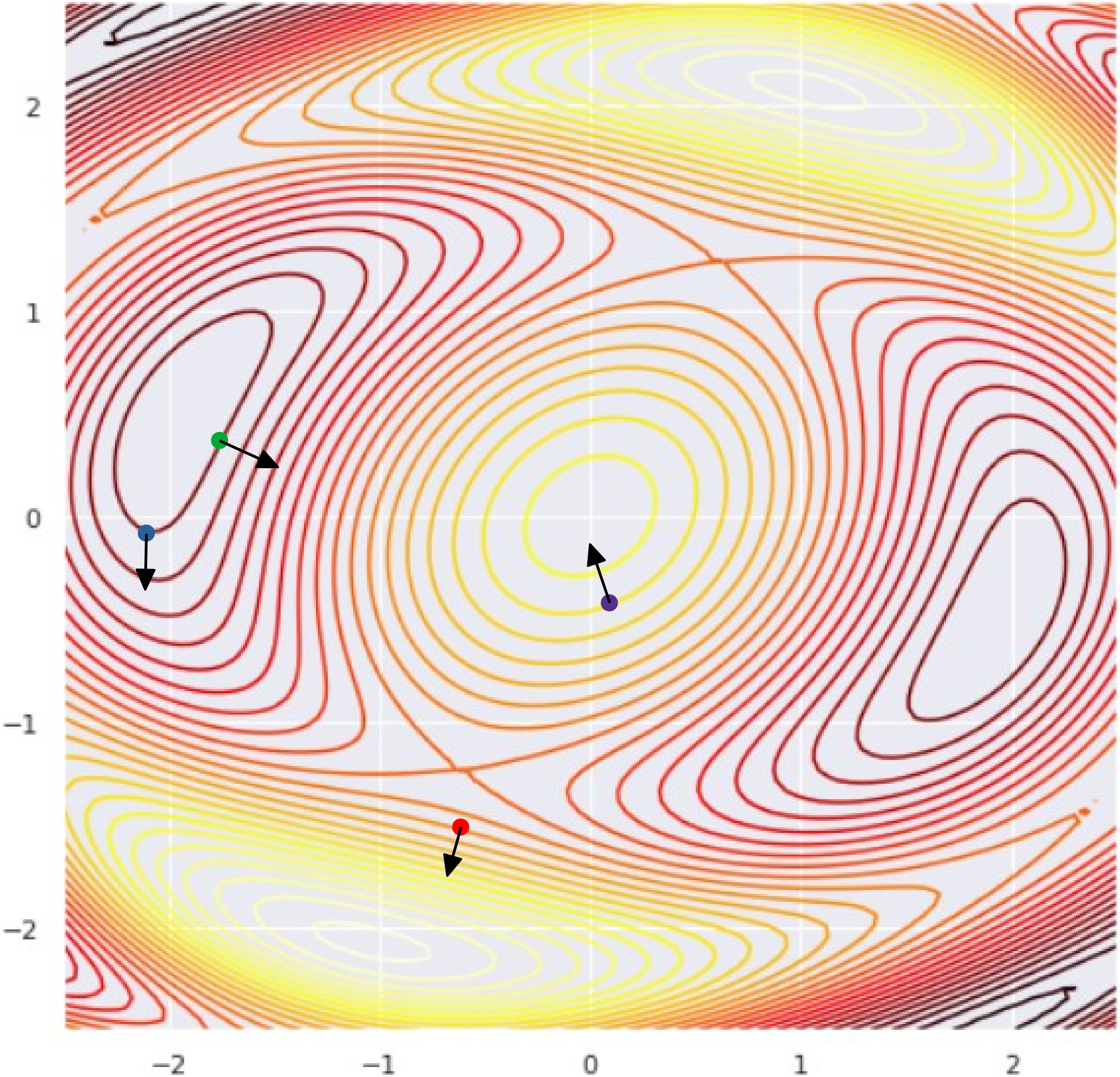
$$\frac{\partial f}{\partial x} = x\cos(\frac{1}{2}x^2 + y^2 + 3) - \sin(x - \frac{1}{2}y)$$

$$\frac{\partial f}{\partial y} = 2y\cos(\frac{1}{2}x^2 + y^2 + 3) + \frac{1}{2}\sin(x - \frac{1}{2}y)$$

Méto



Méto



Regressão linear com gradiente

- O método da regressão linear com gradiente envolve **estimar** o gradiente para um conjunto de parâmetros **w**
- Em seguida nós iremos **atualizar** os parâmetros **w** caminhando no sentido **contrário** do gradiente
  - O gradiente indica o sentido de crescimento da função, mas nós queremos minimizar o erro
- Utilizaremos um hiperparâmetro apropriado para a taxa de aprendizado (η) que irá ponderar o gradiente

Regressão linear com gradiente

- É importante **normalizar os dados**
  - O gradiente fornece um "passo" em cada dimensão
  - Se as dimensões forem muito divergentes, então o caminhamento será dominado pelas dimensões de magnitude mais elevada

Regressão linear com gradiente

- Se o erro do regressor pode ser descrito como $E(\mathbf{w})$

$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}[y(x_i, \mathbf{w}) - t_i]^2$$

- Então seu gradiente é o vetor resultante das derivadas parciais

$$\nabla E = \left(\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \ldots, \frac{\partial E}{\partial w_M}\right)$$

Regressão linear com gradiente

- As derivadas parciais podem ser calculadas como

$$\frac{\partial E(\mathbf{w})}{\partial w_j} = \frac{\partial}{\partial w_j} \frac{1}{2N} \sum_{i=1}^{N} [(y(x_i, \mathbf{w}) - t_i]^2$$

$$\frac{\partial E(\mathbf{w})}{\partial w_j} = \frac{1}{2N} \sum_{i=1}^{N} \frac{\partial}{\partial w_j} [(y(x_i, \mathbf{w}) - t_i]^2$$

$$\frac{\partial E(\mathbf{w})}{\partial w_j} = \frac{1}{2N} \sum_{i=1}^{N} 2\,[y(x_i, \mathbf{w}) - t_i]\, \frac{\partial}{\partial w_j} y(x_i, \mathbf{w}) - t_i$$

Regressão linear com gradiente

- Expandindo a expressão $y(x_i, \mathbf{w})$ - t_i, temos

$$\frac{\partial}{\partial w_j} y(x_i, \mathbf{w}) - t_i = \frac{\partial}{\partial w_j} w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M - t_i$$

- Todos os termos w_k para os quais $k \neq j$ serão tratados como constantes. Por exemplo:

$$\frac{\partial}{\partial w_0} y(x_i, \mathbf{w}) - t_i = \frac{\partial}{\partial w_0} w_0 + \cancelto{0}{w_1 x} + \cancelto{0}{w_2 x^2} + \cancelto{0}{\ldots} + \cancelto{0}{w_M x^M} - \cancelto{0}{t_i}$$

Regressão linear com gradiente

- Expandindo a expressão $y(x_i, \mathbf{w})$ - t_i, temos

$$\frac{\partial}{\partial w_j} y(x_i, \mathbf{w}) - t_i = \frac{\partial}{\partial w_j} w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M - t_i$$

- Todos os termos w_k para os quais $k \neq j$ serão tratados como constantes. Por exemplo:

$$\frac{\partial}{\partial w_1} y(x_i, \mathbf{w}) - t_i = \frac{\partial}{\partial w_1} \cancelto{0}{w_0} + w_1 x + \cancelto{0}{w_2 x^2} + \cancelto{0}{\ldots} + \cancelto{0}{w_M x^M} - \cancelto{0}{t_i}$$

Regressão linear com gradiente

- Expandindo a expressão $y(x_i, \mathbf{w}) - t_i$, temos

$$\frac{\partial}{\partial w_j} y(x_i, \mathbf{w}) - t_i = \frac{\partial}{\partial w_j} w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M - t_i$$

- Todos os termos w_k para os quais $k \neq j$ serão tratados como constantes. Por exemplo:

$$\frac{\partial}{\partial w_2} y(x_i, \mathbf{w}) - t_i = \frac{\partial}{\partial w_2} \cancelto{0}{w_0} + \cancelto{0}{w_1 x} + w_2 x^2 + \cancelto{0}{\ldots} + \cancelto{0}{w_M x^M} - \cancelto{0}{t_i}$$

Regressão linear com gradiente

- Expandindo a expressão y(x_i, **w**) - t_i, temos

$$\frac{\partial}{\partial w_j} y(x_i, \mathbf{w}) - t_i = \frac{\partial}{\partial w_j} w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M - t_i$$

$$\frac{\partial}{\partial w_j} y(x_i, \mathbf{w}) - t_i = x_j^j$$

Regressão linear com gradiente

- Em suma:

$$\frac{\partial E(\mathbf{w})}{\partial w_j} = \frac{1}{N} \sum_{i=1}^{N} \left[y(x_i, \mathbf{w}) - t_i \right] x_i^j$$

Regressão linear com gradiente

- **Algoritmo** GradienteDescendente($\mathbf{X}$, y, $\mathbf{t}$, M, η, ε, max)

  N ← número de instâncias

  $\mathbf{w}$ ← vetor com M + 1 valores aleatórios em [0, 0.1]

  enquanto $E(\mathbf{w}) > \varepsilon$ **ou** após max iterações

  $\Delta\mathbf{w}$ ← [0, 0, ..., 0] *//M + 1 fatores*

  para i **de** 1 **até** N

  para j **de** 0 **até** M

  $\Delta w_j \leftarrow \Delta w_j + \eta(y(x_i, \mathbf{w}) - t_i)\cdot x_i$^j

  $\mathbf{w} \leftarrow \mathbf{w} - \Delta\mathbf{w}/N$

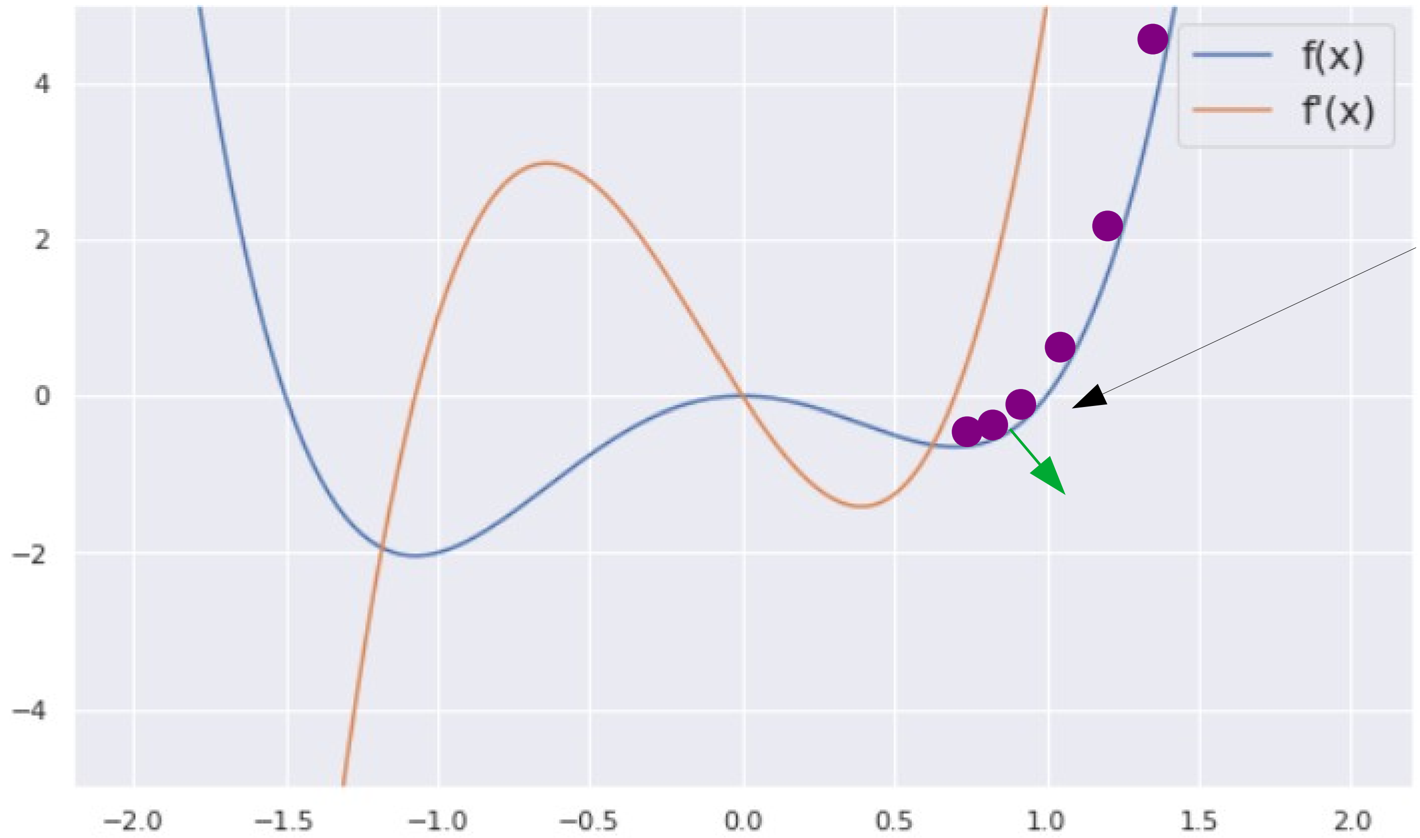
  retorne $\mathbf{w}$

Momento

- A função de perda da regressão linear possui um único mínimo global, portanto o algoritmo sempre irá convergir se o valor de η for escolhido adequadamente
  - Uma abordagem comum é ir reduzindo o valor de η conforme o algoritmo avança em número de iterações
- No caso geral, entretanto, as funções de perda podem conter mínimos locais
  - Nesse caso podemos usar o gradiente descendente com momento

Momento

- O momento leva em consideração o valor do gradiente na iteração anterior

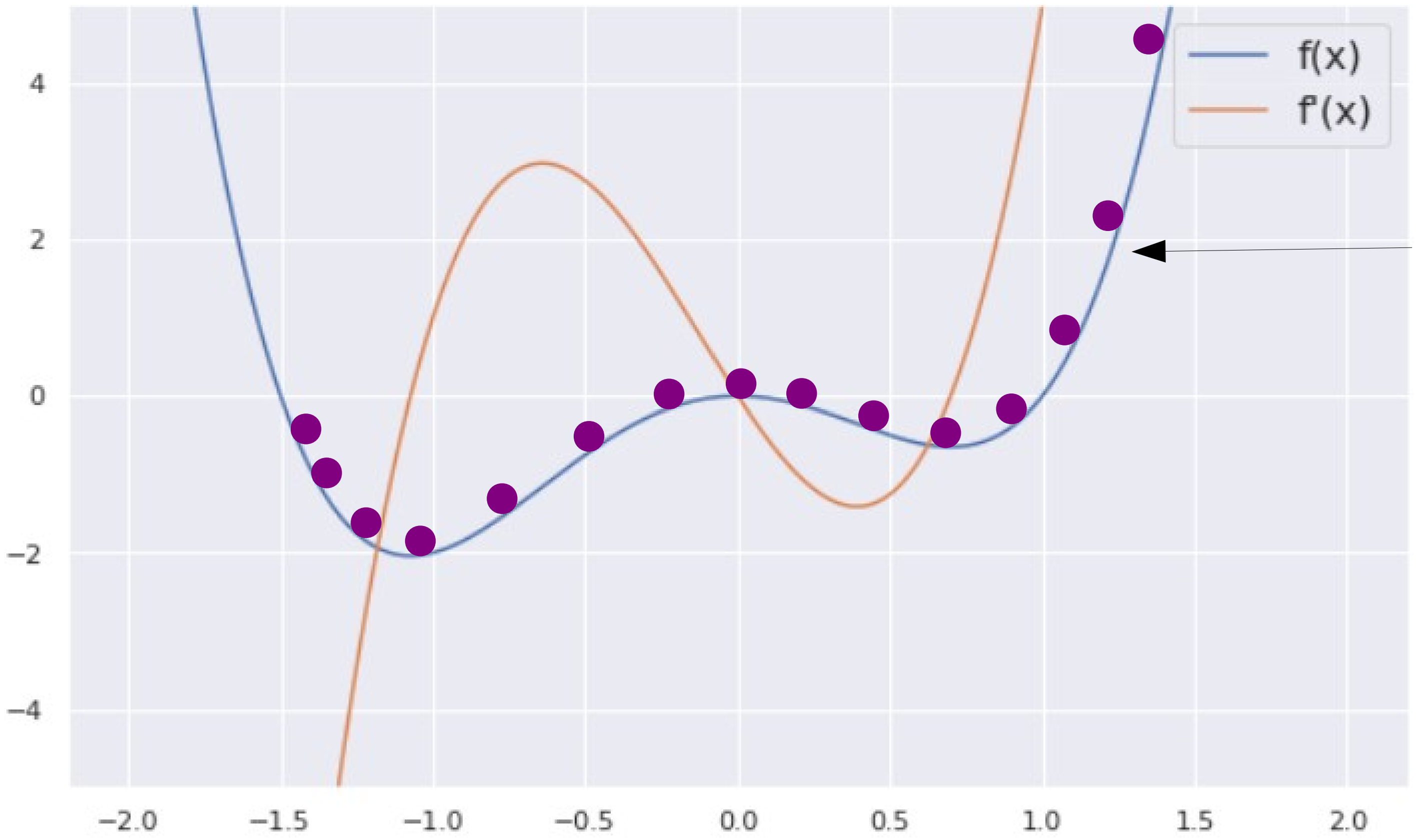


O valor do gradiente aqui é bastante reduzido

Possivelmente ficaremos "presos" nesse mínimo local

Momento

- O momento leva em consideração o valor do gradiente na iteração anterior



O momento simula a "inércia" com que "descemos" esse trecho do espaço de busca

- **Algoritmo** GD_Momento($\mathbf{X}$, y, $\mathbf{t}$, M, η, ε, μ, max)

  N ← número de instâncias

  $\mathbf{w}$ ← vetor com M + 1 valores aleatórios em [0, 0.1]

  $\mathbf{w}_{prev}$ ← [0, 0, ..., 0] *//M + 1 fatores*

  enquanto $E(\mathbf{w})$ > ε **ou** após max iterações

      Δ$\mathbf{w}$ ← [0, 0, ..., 0] *//M + 1 fatores*

      para i **de** 1 **até** N

          para j **de** 0 **até** M

              $\Delta w_j \leftarrow \Delta w_j + \eta(y(\mathrm{x}_i, \mathbf{w}) - t_i)\cdot\mathrm{x}_i$^j

      $\mathbf{w} \leftarrow \mathbf{w} + \mu\cdot\mathbf{w}_{prev} - \Delta\mathbf{w}/N$

      $\mathbf{w}_{prev} \leftarrow \mathbf{w}$

  retorne $\mathbf{w}$

Gradiente descendente estocástico

- Outra forma de evitar mínimos locais é o **gradiente descendente estocástico**
- Em vez de calcular o gradiente para todos os exemplos e atualizar ao final, calculamos o gradiente para cada exemplo
- Não seguimos um único gradiente $E(\mathbf{w})$ para todos os exemplos e sim diversos $E(\mathbf{w})$ para cada exemplo

Gradiente descendente estocástico

- **Algoritmo** GD_Estocástico($\mathbf{X}$, y, $\mathbf{t}$, M, η, ε, `max`)

  N ← número de instâncias

  $\mathbf{w}$ ← vetor com M + 1 valores aleatórios em [0, 0.1]

  enquanto $E(\mathbf{w}) > \varepsilon$ **ou** após `max` iterações

  para i **de** 1 **até** N

  $\Delta\mathbf{w} \leftarrow [0, 0, \ldots, 0]$ *//M + 1 fatores*

  para j **de** 0 **até** M

  $\Delta w_j \leftarrow \Delta w_j + \eta(y(\mathrm{x}_i, \mathbf{w}) - t_i)\cdot \mathrm{x}_i$^j

  $\mathbf{w} \leftarrow \mathbf{w} - \Delta\mathbf{w}/\mathrm{N}$

  retorne $\mathbf{w}$