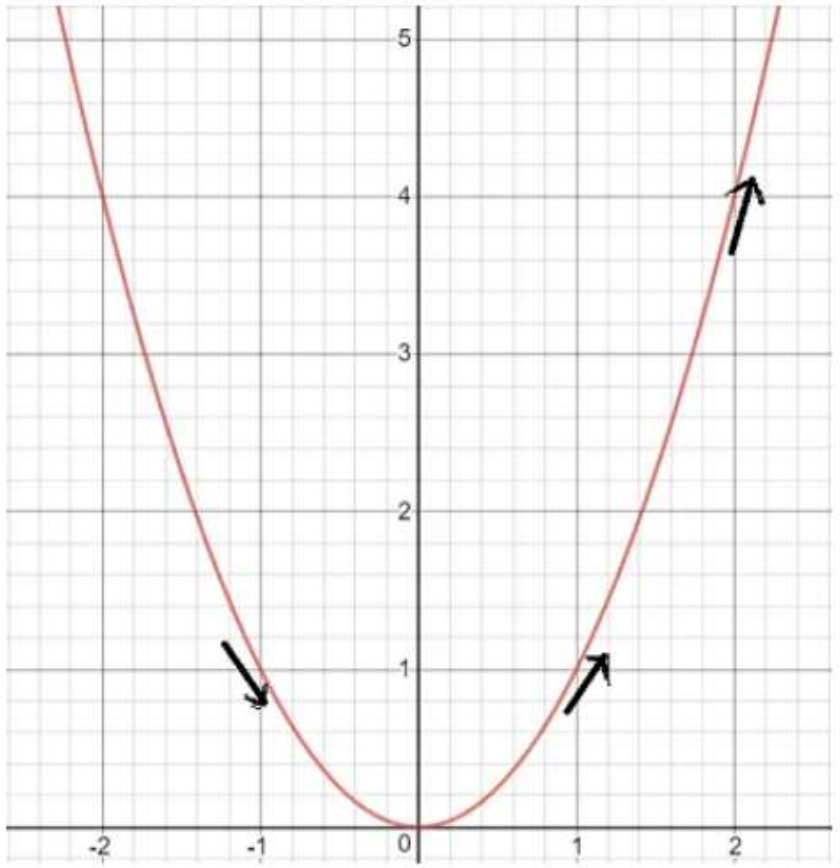
CSE: Faculty of Computer Science and Engineering
Thuyloi University

Gradient Descent

Trình bày: PGS.TS Nguyễn Hữu Quỳnh

Giới thiệu

Ý nghĩa của đạo hàm: đạo là con đường, hàm là hàm số nên đạo hàm chỉ sự biến đổi của hàm số hay có tên là độ dốc của đồ thị.



Đồ thị hàm số $y = x^2$

Giới thiệu

- Như chúng ta biết, đạo hàm $\mathrm{f(x)} = x^2$ là $f'(x) = \frac{df(x)}{dx} = 2x$
- Nhận xét:
  - $f'(1) = 2 < f'(2) = 4$ do đó đồ thị tại điểm $\mathrm{x} = 2$ dốc hơn tại điểm $\mathrm{x} = 1$ => giá trị tuyệt đối của đạo hàm tại một điểm càng lớn thì tại điểm đó đồ thị tại điểm đó càng dốc
  - $f'(-1) = -2 < 0$ => khi x tăng thì $f(x)$ giảm và ngược lại

Thuật toán Gradient descent

Gradient descent là thuật toán tìm giá trị nhỏ nhất của hàm số f(x) dựa trên đạo hàm.

Bước 1: Khởi tạo giá trị $x = x_0$ tùy ý

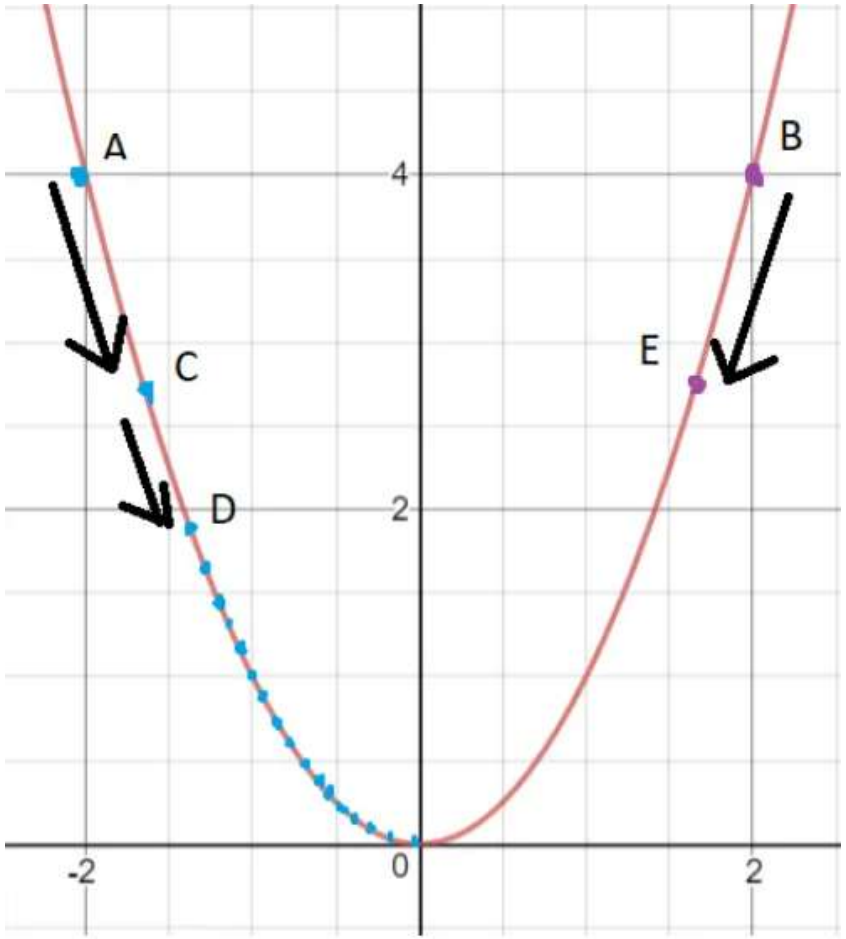
Bước 2: Gán x = x – learning_rate * f'(x) (learning_rate là hằng số không âm ví dụ learning_rate = 0.001)

Bước 3: Tính lại f(x):

- Nếu f(x) đủ nhỏ thì dừng lại.
- Ngược lại tiếp tục bước 2

Thuật toán Gradient descent

Ví dụ: tìm giá trị nhỏ nhất hàm $y = x^2$



Thuật toán Gradient descent

Bước 1: Khởi tạo giá trị ngẫu nhiên x = -2 (điểm A).

Bước 2:

-Khởi tạo tại A: Do tại A đồ thị giảm nên f'(x=-2) = 2*(-2) = -4 < 0 => Khi gán x = x – learning_rate * f'(x) nên x tăng nên đồ thị bước tiếp theo ở điểm C. Tiếp tục thực hiện bước 2, gán x = x – learning_rate * f'(x) thì đồ thị ở điểm D,… => hàm số giảm dần dần tiến tới giá trị nhỏ nhất.

-Khởi tạo tại B: x = 2 thì đạo hàm tại B dương nên do x = x – learning_rate * f'(x) giảm -> đồ thị ở điểm E -> rồi tiếp tục gán x=x -learning_rate * f'(x) thì hàm f(x) cũng sẽ giảm dần dần đến giá trị nhỏ nhất.

Thuật toán Gradient descent

Nhận xét:

- Thuật toán hoạt động rất tốt trong trường hợp không thể tìm giá trị nhỏ nhất bằng đại số tuyến tính.
- Việc quan trọng nhất của thuật toán là tính đạo hàm của hàm số theo từng biến sau đó lặp lại bước 2.

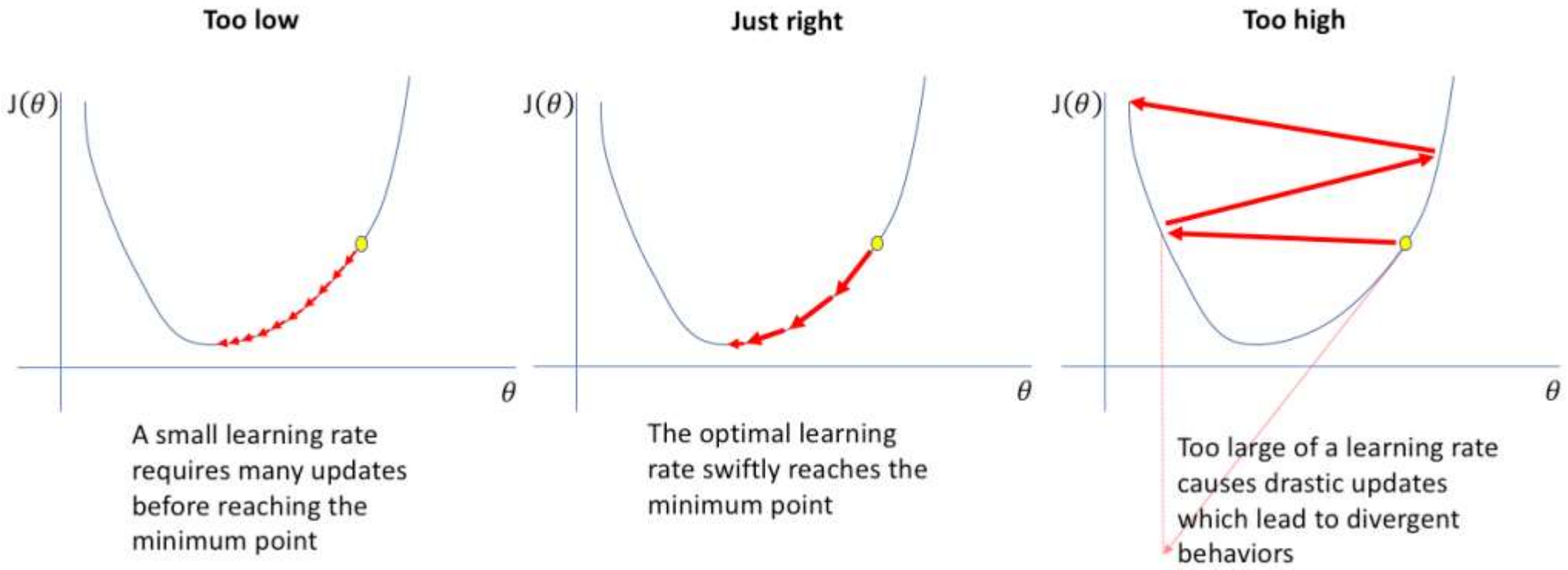
Thuật toán Gradient descent

Việc chọn hệ số learning_rate cực kì quan trọng, có 3 trường hợp:

- Nếu learning_rate nhỏ: mỗi lần hàm số giảm rất ít nên cần rất nhiều lần thực hiện bước 2 để hàm số đạt giá trị nhỏ nhất
- Nếu learning_rate hợp lý: sau một số lần lặp bước 2 vừa phải thì hàm sẽ đạt giá trị đủ nhỏ.
- Nếu learning_rate quá lớn: sẽ gây hiện tượng overshoot và không bao giờ đạt được giá trị nhỏ nhất của hàm.

Thuật toán Gradient descent

- 3 giá trị learning_rate



Thuật toán Gradient descent

- Áp dụng vào bài toán tìm giá trị nhỏ nhất của hàm

$$J(w_0, w_1) = \frac{1}{2} * (\sum_{i=1}^{N} (\hat{y}_i - y_i)^2) = \frac{1}{2} * (\sum_{i=1}^{N} (w_0 + w_1 * x_i - y_i)^2)$$

- Việc tìm giá trị lớn nhất hàm này hoàn toàn có thể giải được bằng đại số nhưng để giới thiệu thuật toán Gradient descent

- Việc quan trọng nhất của thuật toán gradient descent là tính đạo hàm của hàm số nên giờ ta sẽ đi tính đạo hàm theo từng biến.

- Nhắc lại kiến thức h'(x) = f(g(x))’ = f'(g)*g'(x). Ví dụ:

Nhắc lại kiến thức h'(x) = f(g(x))’ = f'(g)*g'(x). Ví dụ:

$h(x) = (3x + 1)^2$ thì $f(x) = x^2, g(x) = 3x + 1 => h'(x) = f'(g) * g'(x) = f'(3x + 1) * g'(x) = 2 * (3x + 1) * 3 = 6 * (3x + 1).$

Thuật toán Gradient descent

Tại 1 điểm (x_i, y_i) gọi

$$f(w_0, w_1) = \frac{1}{2} * (w_0 + w_1 * x_i - y_i)^2$$

Ta có:

$$\frac{df}{dw_0} = w_0 + w_1 * x_i - y_i$$

$$\frac{df}{dw_1} = x_i * (w_0 + w_1 * x_i - y_i)$$

Do đó

$$\frac{dJ}{dw_0} = \sum_{i=1}^{N} (w_0 + w_1 * x_i - y_i)$$

$$\frac{dJ}{dw_1} = \sum_{i=1}^{N} x_i * (w_0 + w_1 * x_i - y_i)$$

CSE: Faculty of Computer Science and Engineering
Thuyloi University

Theo tiếp cận đại số tuyến tính

Ví dụ với Python

- Xét hàm số $f(x)=x^2+5\sin(x)$ với đạo hàm $f'(x)=2x+5\cos(x)$
- GS bắt đầu từ một điểm x_0 , tại vòng lặp thứ t, ta sẽ cập nhật như sau:

$$x_{t+1} = x_t - \eta(2x_t+5\cos(x_t))$$

- Điểm khởi tạo khác nhau: $x_0=4.5$

Tối ưu hàm mất mát của Linear Regression bằng GD

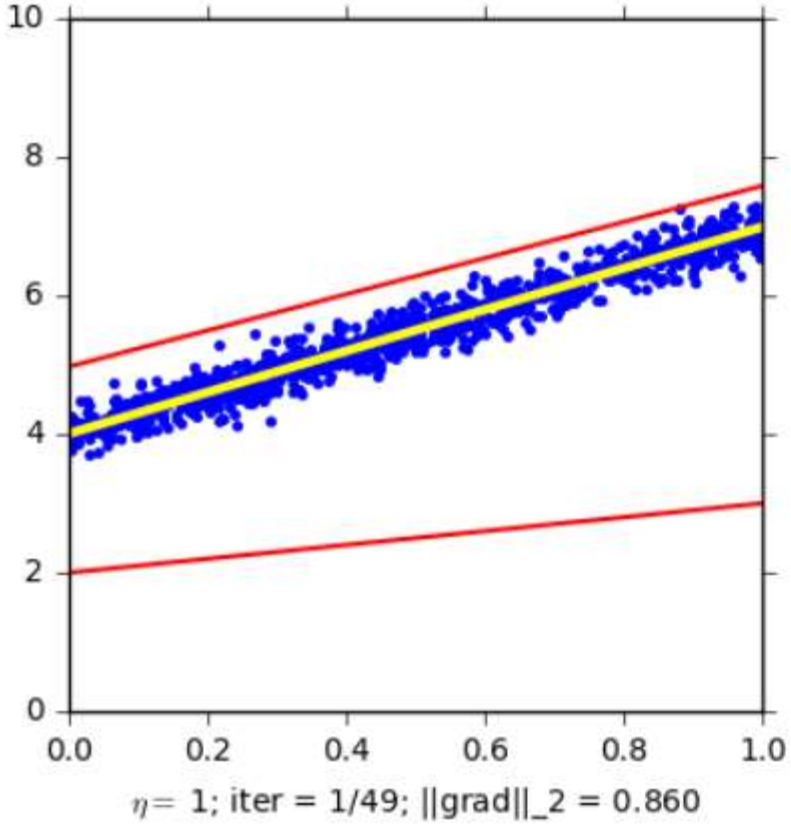
- Hàm mất mát của Linear Regression là:

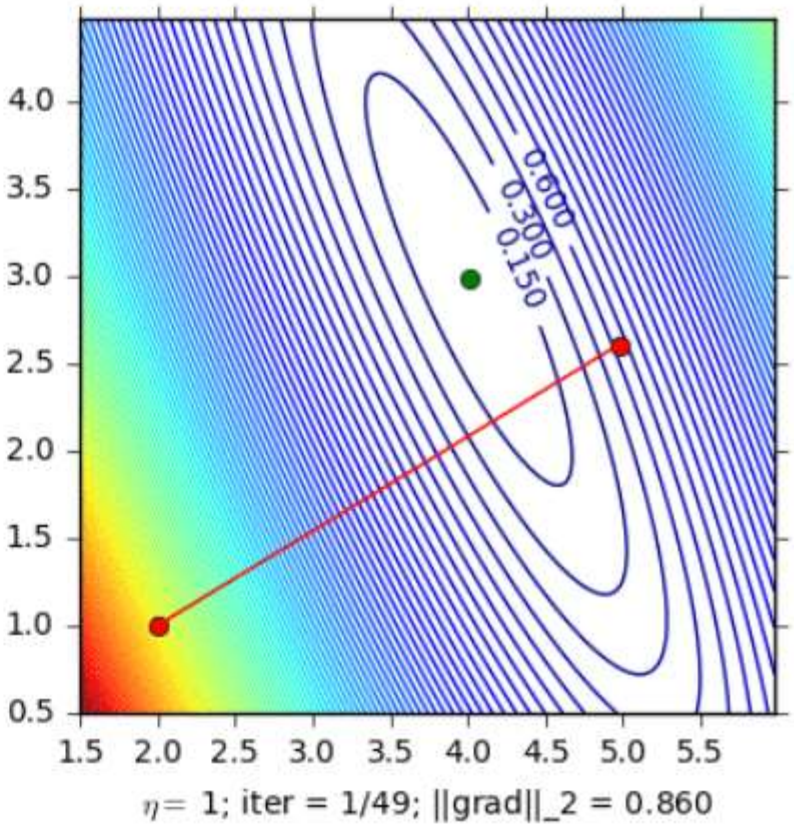
$$\mathcal{L}(w) = \frac{1}{2N} \|\mathrm{y} - \mathrm{Xw}\|_2^2$$

- Đạo hàm của hàm mất mát là:

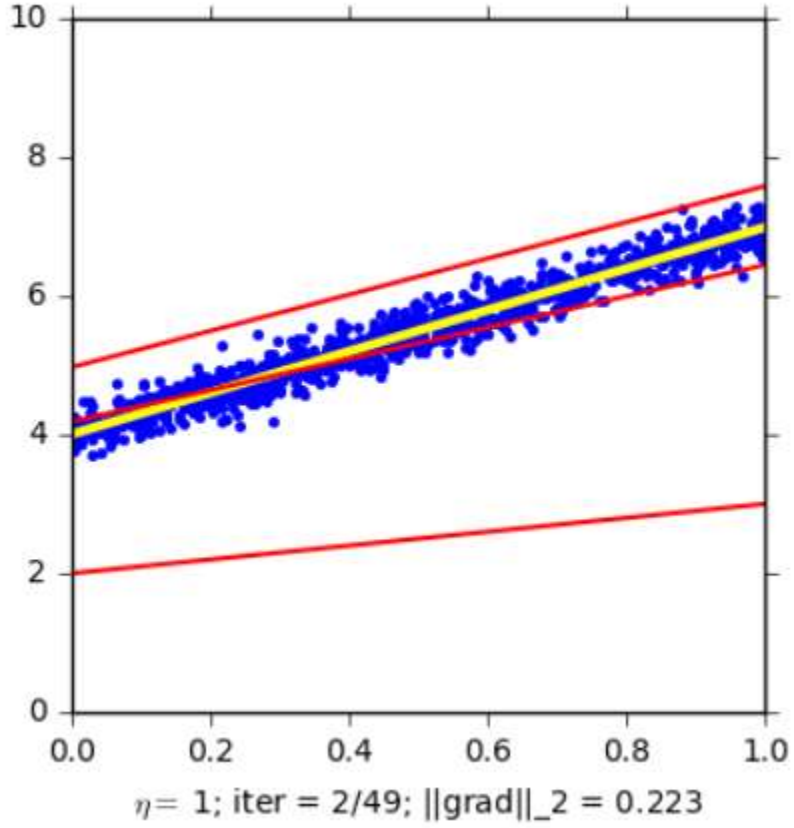
$$\nabla_w \mathcal{L}(w) = \frac{1}{N} \mathrm{X}^T (\mathrm{Xw} - \mathrm{y})$$

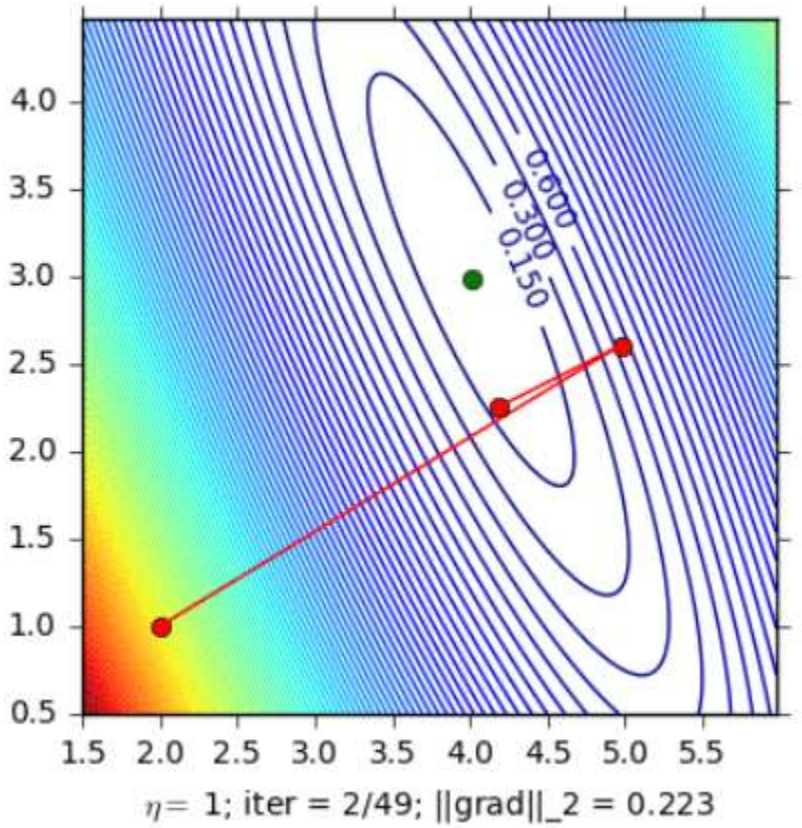
Tối ưu hàm mất mát của Linear Regression bằng GD





Tối ưu hàm mất mát của Linear Regression bằng GD





Tối ưu hàm mất mát của Linear Regression bằng GD