Tài liệu tham khảo

- Vũ Hữu Tiệp, *Machine Learning cơ bản, 2018.* Link download https://github.com/tiepvupsu/ebookMLCB
- Blog: https:// machinelearningcoban.com
- Facebook Page: https:// www.facebook.com/machinelearningbasicvn/
- Facebook Group: https:// www.facebook.com/ groups/machinelearningcoban/
- Interactive Learning: https:fundaml.com
- Bài giảng Học máy của PGS.TS Nguyễn Hữu Quỳnh
- Bài giảng Học máy của PGS.TS Nguyễn Thanh Tùng
- Bài giảng Học máy của TS Nguyễn Thị Kim Ngân

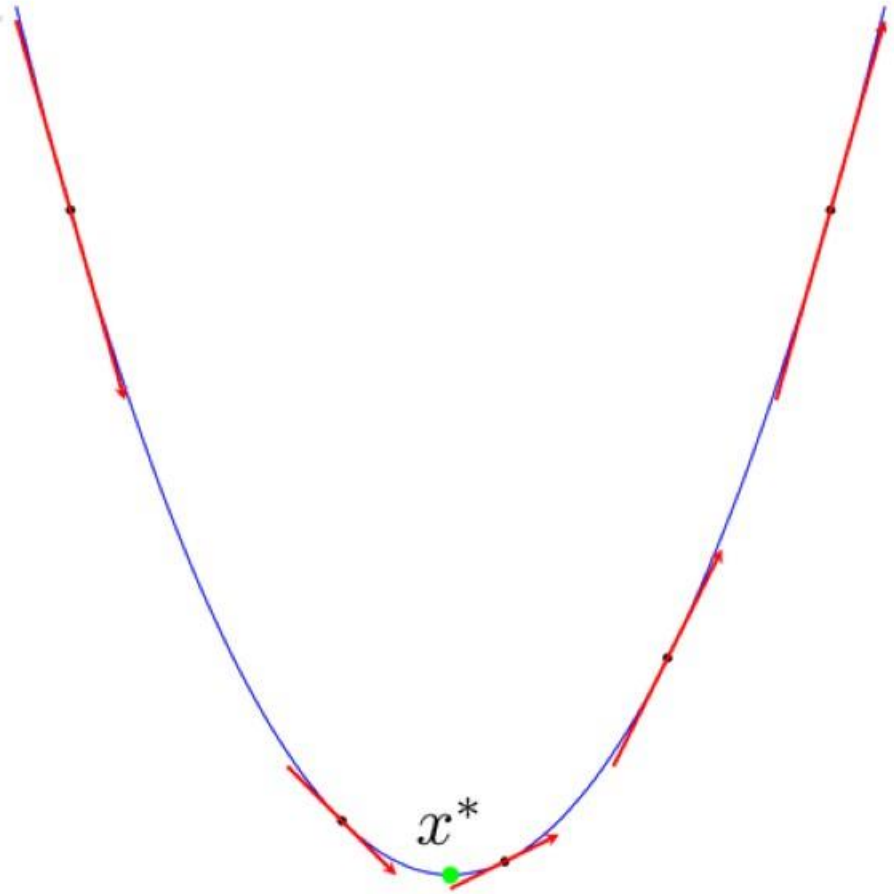
CSE: Faculty of Computer Science and Engineering
Thuyloi University

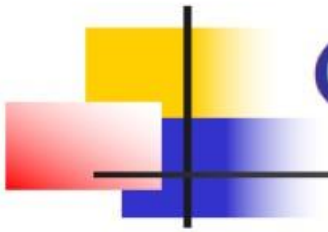
Gradient Descent

Giới thiệu

$$f(x) = \tfrac{1}{2}(x-1)^2 - 2$$

x	$-\infty$		1		$+\infty$
$f'(x)$		$-$	0	$+$	
$f(x)$	$+\infty$	$\searrow$	-2	$\nearrow$	$+\infty$





Giới thiệu

- Local minimum (điểm cực tiểu): là điểm tại đó đạo hàm của hàm số bằng 0
- Global minimum: là điểm mà tại đó hàm số đạt giá trị nhỏ nhất

Giới thiệu

Xét hàm số một biến f(x) có đạo hàm mọi nơi:

- Điểm local minimum x^* của hàm số là điểm có đạo hàm $f'(x^*)$ bằng 0. Trong lân cận của x^*:
  - Đạo hàm của các điểm phía bên trái x^* là âm
  - Đạo hàm của các điểm phía bên phải x^* là dương
- Đường tiếp tuyến với đồ thị hàm số tại x_t có hệ số góc bằng đạo hàm của hàm số tại x_t

Gradient Descent

Trong Machine learning, ta thường xuyên phải tìm giá trị nhỏ nhất của một hàm số

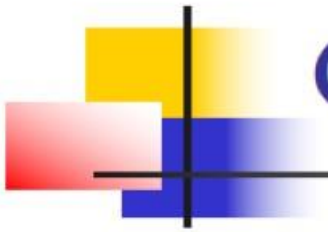
- Các điểm cực tiểu là nghiệm của phương trình đạo hàm bằng 0
- Nếu tìm được toàn bộ (hữu hạn) các điểm cực tiểu , ta chỉ cần thay từng điểm local minimum đó vào hàm số rồi tìm điểm làm cho hàm có giá trị nhỏ nhất

Gradient Descent

Việc giải phương trình đạo hàm bằng không thường rất phức tạp hoặc có thể ra vô số nghiệm. Vì:

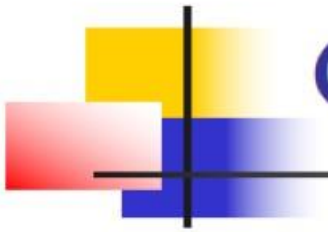
- Sự phức tạp của dạng của đạo hàm
- Các điểm dữ liệu có số chiều lớn
- Có quá nhiều điểm dữ liệu

=>Do đó, **việc tìm global minimum của hàm số có thể bất khả thi**



Gradient Descent

- Người ta thường **cố gắng tìm các điểm local minimum**, và ở một mức độ nào đó, **coi đó là nghiệm cần tìm của bài toán**
- Thực tế cho thấy, trong nhiều bài toán machine learning, các nghiệm local minimum thường đã cho kết quả tốt, đặc biệt là trong neural networks



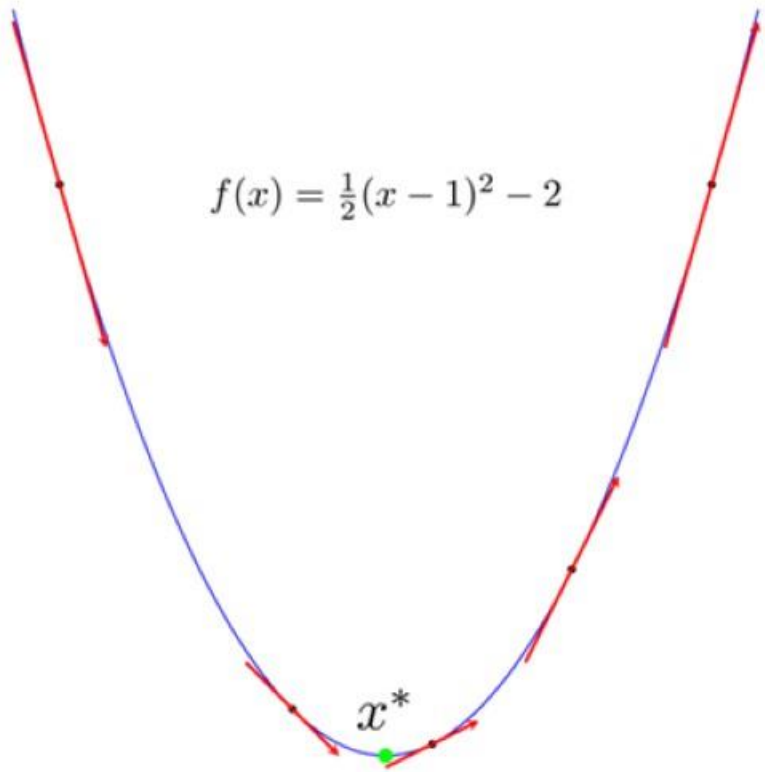
Gradient Descent

Gradient Descent là một trong những phương pháp được dùng nhiều để tìm điểm cực tiểu của bài toán tối ưu:

- Xuất phát từ một điểm được coi là *gần* với nghiệm của bài toán
- Dùng một phép toán lặp để *tiến dần* đến điểm cần tìm (có đạo hàm gần với 0)

Gradient Descent cho hàm 1 biến

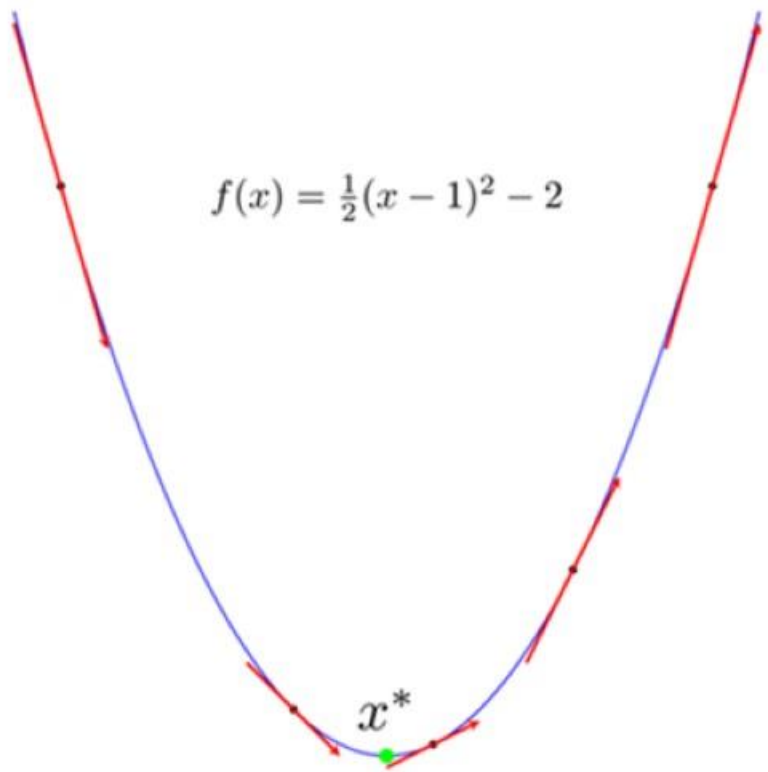
Giả sử x_t là điểm ta tìm được sau vòng lặp thứ t. Ta cần tìm một thuật toán để đưa x_t về càng gần x^* càng tốt.



Gradient Descent cho hàm 1 biến

- Nếu $f'(x_t) > 0$ thì x_t nằm phía phải so với x^* và ngược lại
- x_t càng xa x^* về phía phải thì $f'(x_t)$ càng lớn hơn 0 (và ngược lại)
- Để điểm tiếp theo x_{t+1} gần với x^* hơn, ta cần di chuyển x_t về phía ngược dấu đạo hàm

$$x_{t+1} = x_t - f'(x_t)$$

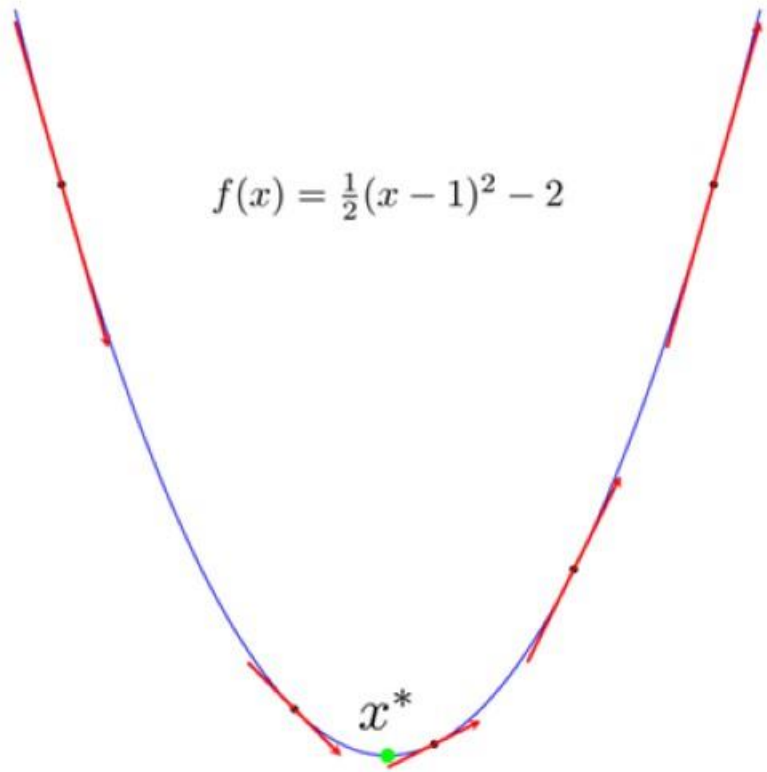


Gradient Descent cho hàm 1 biến

Công thức tổng quát

$$x_{t+1} = x_t - \eta f'(x_t)$$

Trong đó η là một số dương được gọi là tốc độ học (learning rate)



Ví dụ

Xét hàm số $f(x)=x^2+5\sin(x)$

- $f'(x)=2x+5\cos(x)$
- Bắt đầu từ một điểm x_0 , tại vòng lặp thứ t:

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

Ví dụ

Xét hàm số $f(x)=x^2+5\sin(x)$

- $f'(x)=2x+5\cos(x)$
- Bắt đầu từ một điểm x_0 , tại vòng lặp thứ t:

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

- **cost** để tính giá trị của hàm số
- **grad** để tính đạo hàm
- **myGD1** là phần chính thực hiện thuật toán GD nêu phía trên. Đầu vào của hàm số này là learning rate và điểm xuất phát. Thuật toán dừng lại khi đạo hàm có độ lớn đủ nhỏ

Ví dụ

- $f(x)=x^2+5\sin(x)$
- $f'(x)=2x+5\cos(x)$
- Bắt đầu từ một điểm x_0 , tại vòng lặp thứ t:

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

```
def grad(x):
    return 2*x+ 5*np.cos(x)

def cost(x):
    return x**2 + 5*np.sin(x)

def myGD1(x0, eta):
    x = [x0]
    for it in range(100):
        x_new = x[-1] - eta*grad(x[-1])
        if abs(grad(x_new)) < 1e-3: # just a small number
            break
        x.append(x_new)
    return (x, it)
```

Ví dụ

Sau khi đã có các hàm cần thiết, chúng ta thử tìm nghiệm với các điểm khởi tạo khác nhau là $x_0 = -5$ và $x_0 = 5$, với cùng learning rate $\eta = 0.1$.

```
(x1, it1) = myGD1(-5, .1)
(x2, it2) = myGD1(5, .1)
print('Solution x1 = %f, cost = %f, after %d iterations'%(x1[-1], cost(x1[-1]), it1))
print('Solution x2 = %f, cost = %f, after %d iterations'%(x2[-1], cost(x2[-1]), it2))
```

Kết quả:

```
Solution x1 = -1.110667, cost = -3.246394, after 11 iterations
Solution x2 = -1.110341, cost = -3.246394, after 29 iterations
```

Ví dụ

Bài 1. Tìm giá trị cực tiểu của hàm số $f(x)=x^2-2$

Bài 2. Tìm giá trị cực tiểu của hàm số $g(x)=(1/3)x^3-x$

$(a.x^n)'=n.(a.x^{n-1})$

$g'(x)=3.(1/3).x^2-1.x^0=x^2-1$

Gradient Descent cho hàm nhiều biến

Cần global minimum cho hàm f(w), trong đó w là một vector

Đạo hàm của hàm số đó tại một điểm w bất kỳ được ký hiệu là $\nabla_w f(w)$

Thuật toán GD cho hàm nhiều biến f(w)

- Bắt đầu bằng một điểm khởi tạo w_0
- Qui tắc cập nhật ở vòng lặp thứ t là:

$$w_{t+1} = w_t - \eta \nabla_w f(w)$$

$$\text{hoặc } w = w - \eta \nabla_w f(w)$$

Tối ưu hàm mất mát của Linear Regression bằng GD

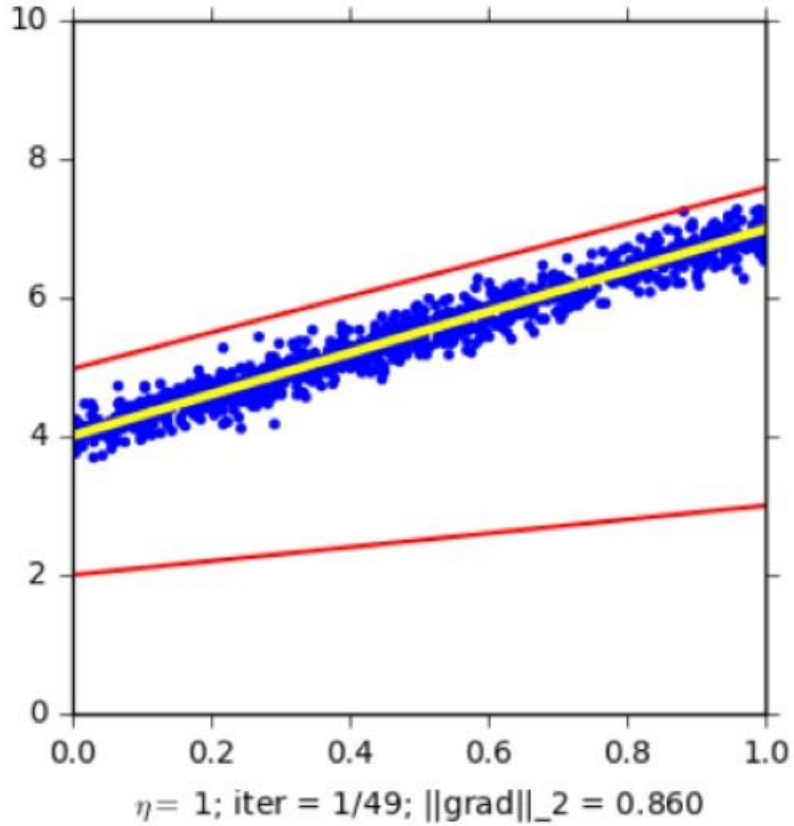
Hàm mất mát của Linear Regression là:

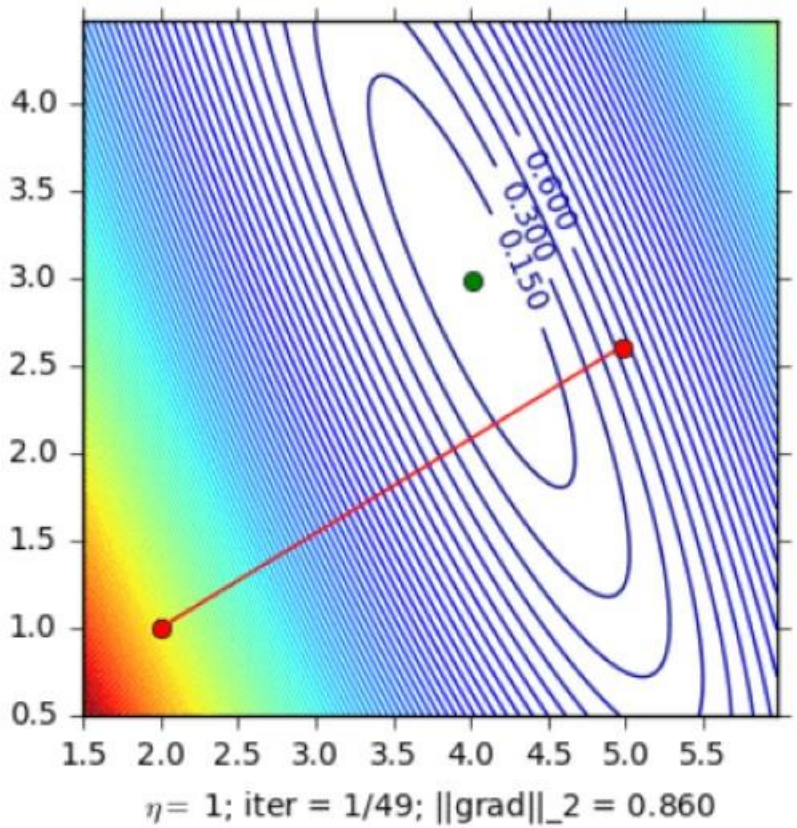
$$\mathcal{L}(w) = \frac{1}{2N} \|\mathrm{y} - \mathrm{Xw}\|_2^2$$

Đạo hàm của hàm mất mát là:

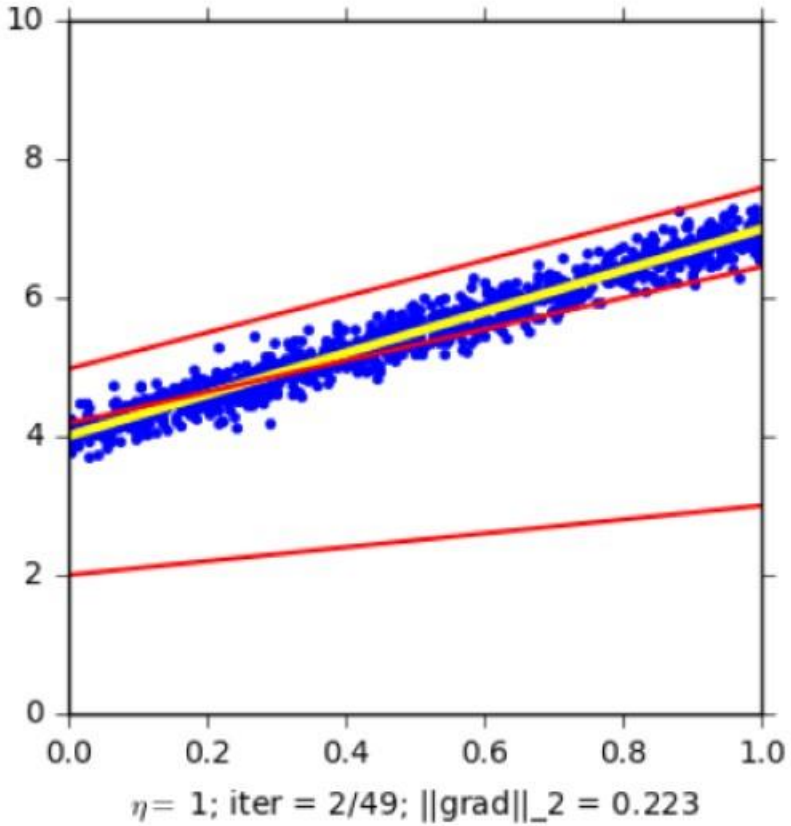
$$\nabla_w \mathcal{L}(w) = \frac{1}{N} \mathrm{X}^T (\mathrm{Xw} - \mathrm{y})$$

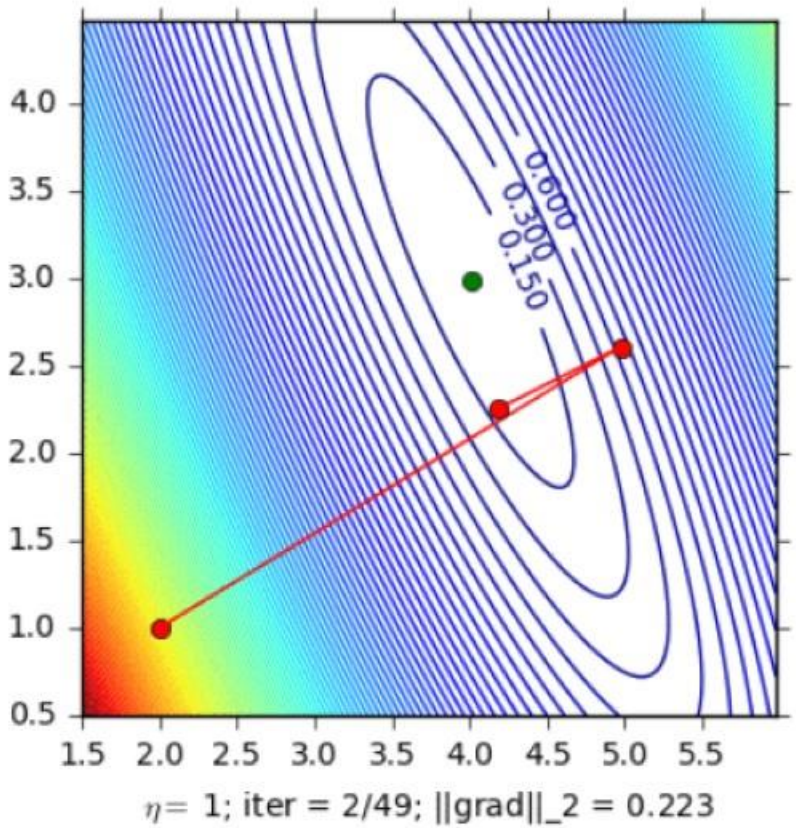
Tối ưu hàm mất mát của Linear Regression bằng GD





Tối ưu hàm mất mát của Linear Regression bằng GD





Tối ưu hàm mất mát của Linear Regression bằng GD