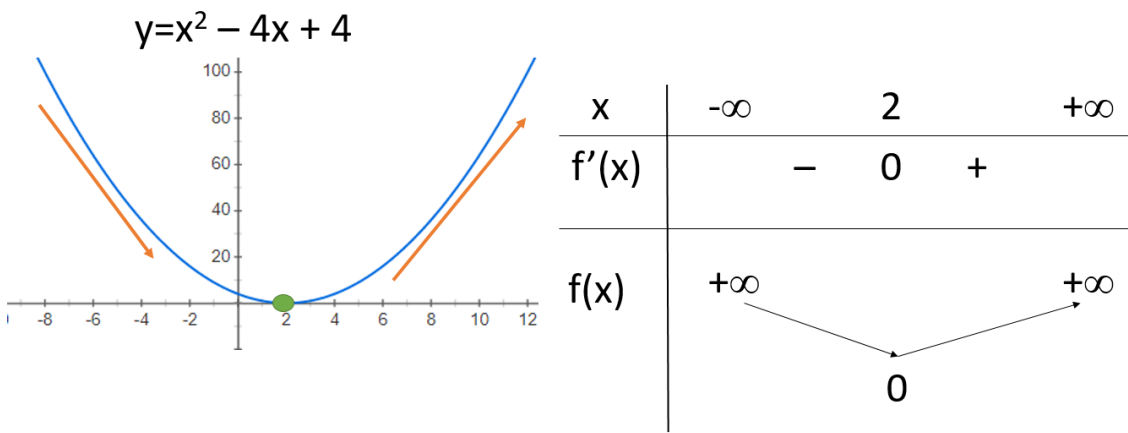


# Gradient descent

## Giới thiệu về gradient descent

Trong học máy, chúng ta thường phải thực hiện các bài toán liên quan đến tìm giá trị nhỏ nhất hoặc lớn nhất của một hàm số bất kỳ. Hàm này được gọi là hàm mất mát, được áp dụng để giúp cho con Robot học một cách chính xác mà con người mong muốn, tức là để tối thiểu hóa lỗi dự đoán. Việc tìm ra giá trị cực tiểu của các hàm mất mát (loss function) trong học máy khá phức tạp, bởi các hàm mất mát này thường có các dạng cực kỳ phức tạp. Do đó, thay vì tìm chính xác điểm cực tiểu, chúng ta sẽ tìm điểm gần bằng (xấp xỉ) điểm cực tiểu.

Xét hàm mất mát là một hàm bậc hai  $y = x^2 - 4x + 4$  có đồ thị là một parabol như hình dưới đây:



Điểm màu xanh ở bên trên được gọi là điểm cực tiểu, và cũng là điểm làm cho hàm số này đạt giá trị nhỏ nhất. Và chắc hẳn các bạn còn nhớ rằng, điểm cực tiểu của một hàm số là vị trí mà đạo hàm của hàm số tại điểm đó có giá trị bằng 0. Các điểm bên tay trái so với điểm cực tiểu có đạo hàm là các giá trị âm, và các điểm bên tay phải có đạo hàm là giá trị dương. Đơn giản hơn, độ lớn của đạo hàm tại các điểm không phải cực tiểu là lớn hơn 0.

Một trong những thuật toán tìm cực tiểu là **Gradient Descent** hay hướng giảm gradient. Thuật toán sẽ xuất phát từ một điểm bất kỳ, sau đó dùng lặp đi lặp lại các bước để sao cho độ lớn của gradient (đạo hàm) tại điểm tiếp theo giảm dần về 0, tức là tiến về điểm cực tiểu của hàm. Ví dụ nếu điểm xuất phát nằm ở nhánh bên phải của parabol, ta sẽ phải đi dần về phía bên trái để đến được điểm cực tiểu. Nói cách khác, ta đã bắt đầu tại một điểm có đạo hàm *dương*, đi dần về phía *âm* (phía điểm có đạo hàm bằng 0). Như vậy, hướng để giảm gradient là ngược hướng với đạo hàm, tức là  $-f'(x)$ .

Trong phạm vi của bài học này, chúng ta sẽ tìm hiểu cụ thể cốt lõi đằng sau thuật toán gradient descent và thực hành cài đặt một ví dụ đơn giản trên Python, tận dụng các kiến thức đã được học để giúp các bạn làm quen với thuật toán này, cũng như luyện tập lại các kỹ năng lập trình đã học, đặc biệt là vận dụng cấu trúc lặp `for` và câu lệnh `break`.

# Gradient Descent cho hàm một biến

Như đã nói ở trên, hướng giảm của gradient là ngược hướng với đạo hàm, suy ra  $\delta\delta$  (đen-ta) có giá trị trái dấu với đạo hàm tại điểm  $x_tx_t$ . Như vậy, về cơ bản,  $\delta\delta$  sẽ tỉ lệ thuận với  $-f'(x_t)$ , ký hiệu tỷ lệ đó là  $\eta\eta$  (ê-ta), được gọi là tỷ lệ học (learning rate). Tỷ lệ học càng lớn thì việc tìm ra điểm giá trị cực tiểu sẽ càng nhanh, nhưng nếu lớn quá, sẽ không thể tìm ra điểm cực tiểu, nhưng tỉ lệ học quá nhỏ thì tìm ra giá trị cực tiểu sẽ rất lâu. Do đó, chúng ta cần tỷ lệ vừa đủ nhỏ để đảm bảo tìm ra được điểm cực tiểu mặc dù thời gian tìm kiếm có thể mất nhiều thời gian hơn. Công thức trên sẽ được viết lại thành

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

Trong đó  $\eta\eta$  là một số dương, dấu trừ thể hiện việc chúng ta phải đi ngược với đạo hàm như đã nói ở trên.

## Bài toán

Để đơn giản, trong phần này, chúng ta sẽ thực hiện việc tối thiểu hóa hàm  $f(x) = x^2 - 4x + 4$  ở ví dụ đầu tiên. Bài toán đặt ra là:

?

Question

Xác định cực tiểu của hàm  $f(x)$  theo thuật toán gradient descent với các tham số điểm khởi tạo ban đầu  $x_0$ , hệ số học learning rate  $\eta$  và số lần lặp tối đa bước được nhập từ bàn phím

Trong đó:

- Điểm xuất phát `x`, hệ số học `eta` và số bước tối đa `step` được nhập từ bàn phím.
- Thuật toán sẽ lặp cho đến khi khoảng cách điểm cực trị và điểm xấp xỉ nhỏ hơn một ngưỡng cho trước là `1e-3`. Điều này là do chúng ta không thể so sánh hai số thực (hãy xem lại bài về các [toán tử so sánh trong Python](#))
- Đến số lần lặp tối đa `step` mà không tìm được điểm cực tiểu (không thoả mãn điều kiện trên) thì thuật toán sẽ dừng lại và kết luận không thể tìm được điểm cực tiểu.

## Định dạng đầu vào, đầu ra.

Đầu vào:

```
1 | 5
2 | 0.1
3 | 100
```

Đầu ra:

```
1 | x_min = 2.003713820117856, loss_function = -0.0037000276579881586, after 30 step
```

Đầu vào:

```
1 | 5
2 | 0.1
3 | 22
```

Đầu ra:

```
1 | Cannot find x_min after 22 step with learning rate eta = 0.1
```