

Phương pháp số trong Khoa học dữ liệu

T.S. Nguyễn Thị Hoài Thương

Trường Đại học Khoa học tự nhiên TP.HCM
Khoa Toán Tin-học
Bộ môn Giải tích

ngththuong@hcmus.edu.vn

Ngày 10 tháng 7 năm 2022

1 Thuật toán Gradient descent

- Đặt vấn đề
- Gradient Descent cho hàm một biến
- Gradient descent cho hàm nhiều biến
- Accelerated gradient descent
- Stochastic gradient descent

Thuật toán Gradient descent

Đặt vấn đề

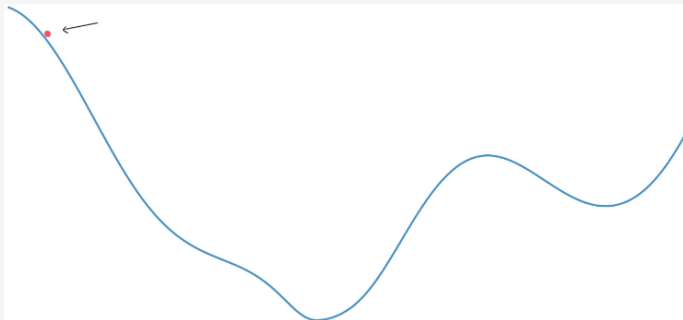


Giả sử bạn đang lạc ở trên một ngọn núi đầy sương mù, tầm nhìn hạn chế. Trời sắp tối và đỉnh núi rất lạnh khi về đêm nên cần phải đi xuống núi để dựng trại. Vì xung quanh toàn là sương mù nên bạn không thể nào xác định chính xác thung lũng ở đâu. Vậy, làm thế nào để bạn có thể xuống được thung lũng một cách nhanh nhất?

Đặt vấn đề

Về mặt toán học:

Giả sử bề mặt của dãy núi được biểu diễn như sau



Phương trình của đồ thị hàm số trên là:

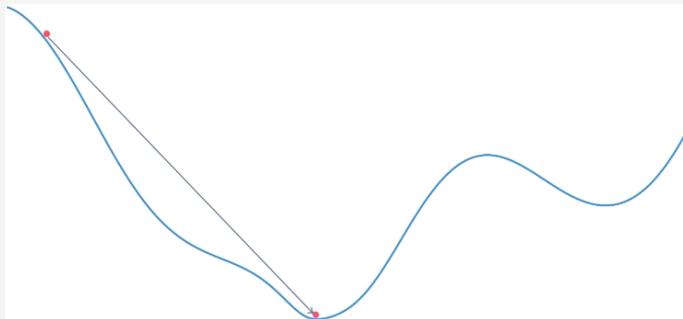
$$f(x) = \frac{\log(|x|^{\sin(x)+2} + 1)}{\log(10)}.$$

Đặt vấn đề

Đạo hàm nó

$$f'(x) = \frac{(\log(\sin(x) + 2) + 1)(\sin(x) + 2)^{\sin(x)+2}}{(|x|^{\sin(x)+2} + 1) \log(10)}.$$

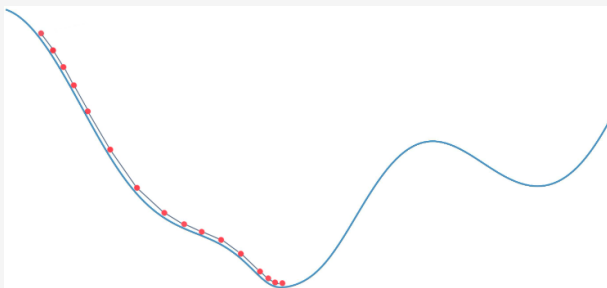
Giải phương trình đạo hàm bằng 0 và tìm được thung lũng gần đó



Đặt vấn đề

Thực tế: Bạn làm như sau

- Bước 1: Khảo sát vị trí đang đứng và tất cả những vị trí xung quanh có thể nhìn thấy, sau đó xác định vị trí nào hướng xuống dốc nhiều nhất.
- Bước 2: Di chuyển đến vị trí đã xác định ở bước 1.
- Bước 3: Lặp lại hai bước trên đến khi nào tới được thung lũng.



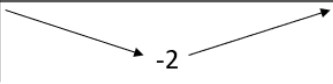
Hay nói cách khác là đi men theo mặt dốc. Đây chính là ý tưởng của thuật toán Gradient Descent.

Gradient Descent cho hàm một biến

Trong kiến thức phổ thông, chúng ta đã biết muốn tìm cực trị của hàm số $y = f(x)$ chúng ta sẽ giải phương trình $f'(x) = 0$.

Ví dụ: Cho $f(x) = \frac{1}{2}(x - 1)^2 - 2$. Ta có

$$f'(x) = x - 1 = 0 \Leftrightarrow x = 1.$$

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

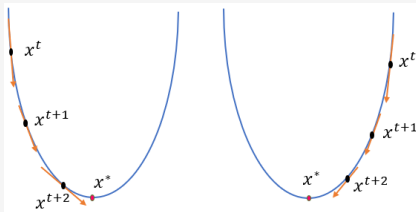
Trong thực tế, không phải lúc nào cũng giải được phương trình $f'(x) = 0$ một cách dễ dàng, có những trường hợp việc giải phương trình này là bất khả thi. Khi đó, chúng ta phải làm sao?

Thuật toán Gradient descent cho chúng ta cách thức tìm các điểm cực tiểu này một cách xấp xỉ sau một số vòng lặp.

Gradient Descent cho hàm một 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.

x	$-\infty$	1	$+\infty$
$f'(x)$	-	0	+
$f(x)$	<div style="display: flex; align-items: center; justify-content: center;"><div style="width: 30%;"></div><div style="text-align: center;">-2</div><div style="width: 30%;"></div></div>		



Hình 1: Bảng biến thiên và đồ thị của hàm $f(x) = \frac{1}{2}(x - 1)^2 - 2$.

Chúng ta thấy rằng:

- Nếu $f'(x_t) > 0$ thì $x(t)$ nằm bên phải so với x^* (và ngược lại). Để điểm tiếp theo x_{t+1} gần với x^* hơn, chúng ta cần di chuyển x_t về phía bên trái, tức là về phía âm. Nói cách khác, **chúng ta cần di chuyển ngược dấu với đạo hàm**.
- Khi x_t càng xa x^* về phía bên phải thì $f'(x_t)$ càng lớn hơn 0 (và ngược lại). Vậy lượng di chuyển từ x_t đến x_{t+1} tỉ lệ thuận với $-f'(x_t)$.

Gradient Descent cho hàm một biến

Do đó, chúng ta có một cách thiết lập đơn giản là

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

trong đó: η là một số dương được gọi là *learning rate*.

Thuật toán:

Đầu vào: Hàm f , learning rate η , giá trị ban đầu x_t , bước lặp lớn nhất N , sai số ϵ

Đầu ra: Giá trị cực tiểu x_{t+1} của hàm f .

- Bước 1: Tính đạo hàm riêng $f'(x)$.
- Bước 2: Đặt $i = 0$.
- Bước 3: while $i \leq N$:
 - ▶ Bước 3.1: Tính $x_{t+1} = x_t - \eta f'(x_t)$
 - ▶ Bước 3.2: Nếu $|f'(x_{t+1})| < \epsilon$ thì
Xuất ra màn hình giá trị x_{t+1}
Dừng lại
 - ▶ Bước 3.3: $x_t = x_{t+1}$, $i = i + 1$
- Bước 4: Xuất ra màn hình thông báo: thuật toán không thành công sau N bước lặp.

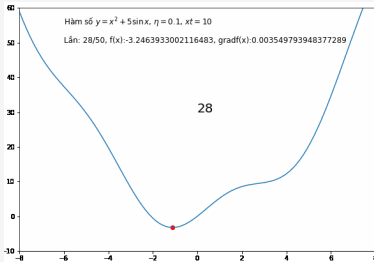
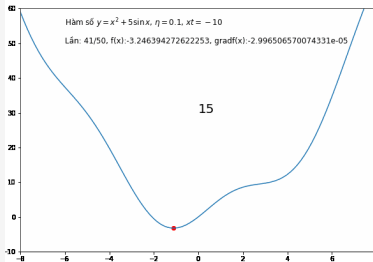
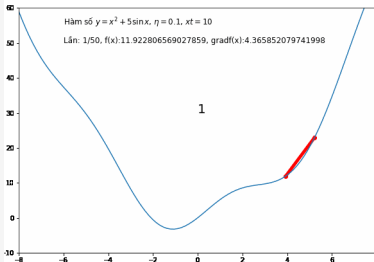
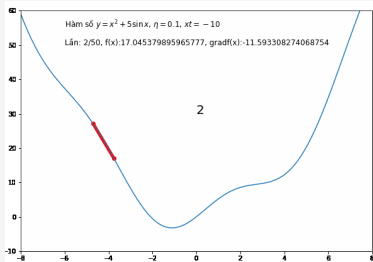
Gradient Descent cho hàm một biến

Bài tập:

- a) Dùng Python để viết hàm thực hiện thuật toán Gradient descent nêu phía trên.
- b) Xét hàm số $f(x) = x^2 + 5 \sin(x)$. Áp dụng hàm đã viết ở câu a) để tìm cực tiểu của hàm f .

Gradient Descent cho hàm một biến

(Cho sinh viên xem hai files hình động 2_10 và 2_11)



Gradient Descent cho hàm một biến

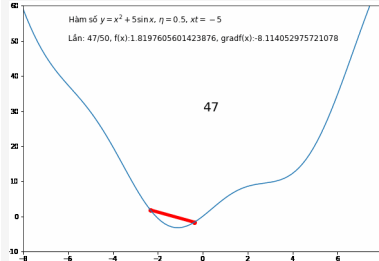
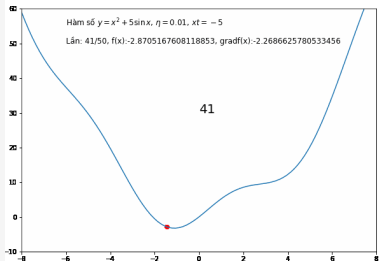
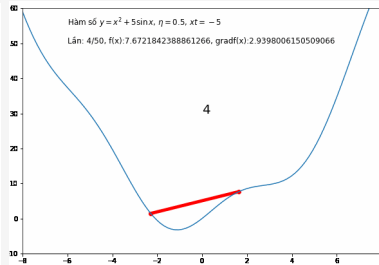
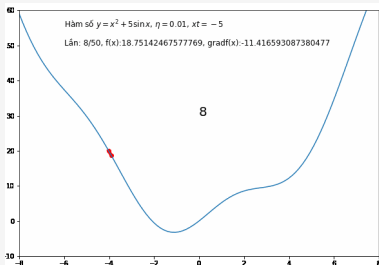
Từ hình minh họa trên ta thấy rằng

- Các hình bên trái tương ứng với $x_t = -10$, nghiệm hội tụ nhanh hơn vì điểm ban đầu x_t gần với điểm cực tiểu -1.11 hơn.
- Hơn thế nữa, với $x_t = 10$ ở hình bên phải, *đường đi* của nghiệm có chứa một khu vực có đạo hàm khá nhỏ gần điểm có hoành độ bằng 3. Điều này khiến cho thuật toán *la cà ở đây khá lâu*. Khi vượt qua được điểm này thì mọi việc diễn ra rất tốt đẹp.

Gradient Descent cho hàm một biến

Learning rate khác nhau

(Cho sinh viên xem hai files hình động 2_16 và 2_17)



Gradient Descent cho hàm một biến

Ta quan sát thấy hai điều:

- Với *learning rate nhỏ* $\eta = 0.01$, tốc độ hội tụ rất chậm. Trong ví dụ này, chúng ta chọn số vòng lặp tối đa là 100 nên thuật toán dừng lại trước khi tới đích, mặc dù đã rất gần. Trong thực tế, khi việc tính toán trở nên phức tạp, *learning rate nhỏ* quá thấp sẽ ảnh hưởng tới tốc độ của thuật toán rất nhiều, thậm chí không bao giờ tới được đích.
- Với *learning rate nhỏ* lớn $\eta = 0.5$ thuật toán tiến rất nhanh tới gần đích. Tuy nhiên thuật toán không hội tụ được vì bước nhảy quá lớn, khiến nó cứ quẩn quanh ở đích.

Gradient descent cho hàm nhiều biến

- Giả sử ta cần tìm cực tiểu cho hàm $f(\mathbf{x})$, trong đó \mathbf{x} là một vectơ.
- Đạo hàm của hàm số đó tại một điểm \mathbf{x} bất kì được kí hiệu là $\nabla f(\mathbf{x})$.
- Tương tự như hàm một biến, thuật toán Gradient descent cho hàm nhiều biến cũng bắt đầu bằng một điểm dự đoán \mathbf{x}_t . Sau đó, chúng ta áp dụng quy tắc cập nhật là

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t), \quad (1.1)$$

trong đó: nếu $\mathbf{x} = (x_1, x_2, \dots, x_n)$ thì

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right).$$

Gradient descent cho hàm nhiều biến

Thuật toán:

Đầu vào: Hàm f , learning rate η , giá trị ban đầu \mathbf{x}_t , bước lặp lớn nhất N , sai số ϵ

Đầu ra: Giá trị cực tiểu \mathbf{x}_{t+1} của hàm f .

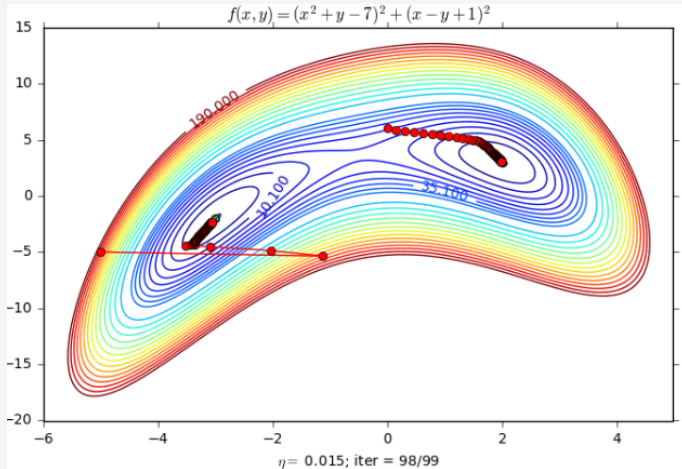
- Bước 1: Tính vectơ gradient $\nabla f(\mathbf{x})$.
- Bước 2: Đặt $i = 0$.
- Bước 3: while $i \leq N$:
 - ▶ Bước 3.1: Tính $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t)$
 - ▶ Bước 3.2: Nếu $\|\nabla f(\mathbf{x}_{t+1})\|_2 < \epsilon$ thì
Xuất ra màn hình giá trị \mathbf{x}_{t+1}
Dừng lại
 - ▶ Bước 3.3: $\mathbf{x}_t = \mathbf{x}_{t+1}$, $i = i + 1$
- Bước 4: Xuất ra màn hình thông báo: thuật toán không thành công sau N bước lặp.

Gradient descent cho hàm nhiều biến

Bài tập:

- a) Dùng Python để viết hàm thực hiện thuật toán Gradient descent nêu phía trên.
- b) Xét hàm số $f(x, y) = (x^2 + y - 7)^2 + (x - y + 1)^2$. Áp dụng hàm đã viết ở câu a) để tìm cực tiểu của hàm f .

Gradient descent cho hàm nhiều biến



Hình 2: Hàm số $f(x, y) = (x^2 + y - 7)^2 + (x - y + 1)^2$ có hai điểm cực tiểu tại $(2, 3)$ và $(-3, -2)$. Trong ví dụ này, tùy vào điểm khởi tạo mà chúng ta thu được các nghiệm cuối cùng khác nhau.

Ưu và nhược điểm của thuật toán Gradient Descent

Ưu điểm:

- Đây là một phương pháp tương đối linh hoạt để tìm cực trị của hàm số bất kỳ.
- Thuật toán Gradient Descent có lợi thế khi việc khảo sát hàm số để tìm cực trị là quá khó hoặc quá phức tạp.

Nhược điểm:

- Ta cần phải lựa chọn giá trị learning rate α phù hợp. Nếu α quá nhỏ thì thuật toán sẽ cần rất nhiều bước để kết thúc. Nếu α quá lớn thì thuật toán có thể không kết thúc được.
- Nếu ta chọn điểm xuất phát mà trong lân cận của nó hàm số không đổi thì Gradient Descent sẽ kết thúc chỉ sau một lần lặp mà không đưa ra được kết quả chính xác. Đây cũng là hạn chế của Gradient Descent.

Accelerated gradient descent

Xét bài toán tối ưu có dạng như sau:

$$\min_{\mathbf{x}} f(\mathbf{x})$$

Nesterov (1983): [Accelerated gradient descent](#)

Để tìm giá trị cực tiểu của hàm $f(\mathbf{x})$, ta giả sử rằng tại thời điểm ban đầu $\mathbf{x}_{-1} = \mathbf{x}_0$. Tại mỗi vòng lặp, ta thực hiện các bước sau:

$$y_k = x_k + \frac{k-1}{k+2}(\mathbf{x}_k - \mathbf{x}_{k-1})$$
$$\mathbf{x}_{k+1} = y_k - \eta \nabla f(y_k).$$

Accelerated gradient descent

Thuật toán:

Đầu vào: Hàm f , learning rate η , giá trị ban đầu \mathbf{x}_0 , bước lặp lớn nhất N , sai số ϵ

Đầu ra: Giá trị cực tiểu của hàm f .

- Bước 1: Tính vectơ gradient $\nabla f(\mathbf{x})$.
- Bước 2: Đặt $i = 0$ và $\mathbf{x}_{i-1} = \mathbf{x}_i$
- Bước 3: while $i \leq N - 1$:
 - ▶ Bước 3.1: Tính

$$y_i = \mathbf{x}_i + \frac{i-1}{i+2} (\mathbf{x}_i - \mathbf{x}_{i-1})$$

$$\mathbf{x}_{i+1} = y_i - \eta \nabla f(y_i)$$

- ▶ Bước 3.2: Nếu $\|\nabla f(\mathbf{x}_{t+1})\|_2 < \epsilon$ thì
Xuất ra màn hình giá trị \mathbf{x}_{t+1}
Dừng lại
 - ▶ Bước 3.3: Cập nhật giá trị $\mathbf{x}_{i-1} = \mathbf{x}_i$, $\mathbf{x}_i = \mathbf{x}_{i+1}$ và $i = i + 1$.
- Bước 4: Xuất ra màn hình thông báo: thuật toán không thành công sau N bước lặp.

Accelerated gradient descent

Bài tập:

- a) Dùng Python để viết hàm thực hiện thuật toán Accelerated gradient descent nêu phía trên.
- b) Xét hàm số $f(x, y) = (x^2 + y - 7)^2 + (x - y + 1)^2$. Áp dụng hàm đã viết ở câu a) để tìm cực tiểu của hàm f .
- c) So sánh kết quả với thuật toán Gradient descent.

Stochastic gradient descent

Xét bài toán tối ưu có dạng như sau:

$$\min_{\mathbf{x}} \frac{1}{m} \sum_{i=1}^m f_i(\mathbf{x}).$$

Đặt

$$f(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m f_i(\mathbf{x}) \Rightarrow \nabla f(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m \nabla f_i(\mathbf{x}).$$

Áp dụng thuật toán Gradient Descent (1.1), ta có giá trị cực tiểu của hàm f được thiết lập như sau

$$\begin{aligned} \mathbf{x}_{t+1} &= \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t) \\ &= \mathbf{x}_t - \eta \times \frac{1}{m} \sum_{i=1}^m \nabla f_i(\mathbf{x}_t). \end{aligned}$$

Stochastic gradient descent

Chúng ta thấy rằng:

- Khi sử dụng thuật toán Gradient descent, chúng ta cần tính đạo hàm tất cả các điểm sau đó lấy trung bình đạo hàm để cập nhật giá trị mới cho x .
- Đối với tập dữ liệu gồm m mẫu thì để việc thực hiện tính đạo hàm trên tất cả mẫu thì độ phức tạp là $O(n)$.

Do đó, với tập dữ liệu lớn, chi phí thực hiện đạo hàm cho toàn bộ tập dữ liệu sẽ rất lớn.

⇒ Chúng ta cần tìm một thuật toán để khắc phục nhược điểm trên.

Stochastic gradient descent

Với thuật toán **Stochastic gradient descent**, ta xem $\nabla f_i(\mathbf{x})$ là một xấp xỉ của mean của các gradient của f_i . Do đó, ta có công thức cập nhật cho \mathbf{x} sẽ là

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \nabla f_{i_t}(x_t) \quad (1.2)$$

trong đó: $i_t \in \{1, 2, \dots, m\}$ là index được chọn ngẫu nhiên ở mỗi lần lặp. Do đó, $\nabla f_{i_t}(\mathbf{x})$ chính là một mẫu ngẫu nhiên của gradient.

Lưu ý: Vì ở mỗi lần lặp thuật toán Stochastic gradient descent chỉ liên quan đến một hàm $\nabla f_{i_t}(x_t)$ nên độ phức tạp của mỗi lần cập nhật giá trị x_{t+1} sẽ thấp hơn hẳn so với thuật toán Gradient descent và độc lập với độ lớn của m .

Do đó, phương pháp này có ưu điểm là chi phí tính toán thấp, có thể tìm được điểm tiệm cận giá trị cực trị và phù hợp với các mô hình cần sự đào tạo nhanh và với lượng mẫu dữ liệu lớn.

Stochastic gradient descent

Ví dụ: Ta xét bài toán dự đoán huyết áp tâm thu của một người dựa vào độ tuổi của họ như sau:

- Giả sử ta có số liệu thống kê về độ tuổi a_i và huyết áp tâm thu b_i của m người lần lượt là

$$(a_1, b_1), \dots, (a_m, b_m).$$

- Yêu cầu đặt ra là dựa vào các dữ liệu trên hãy tìm ra mối liên hệ tuyến tính giữa huyết áp tâm thu và tuổi. Từ đó, ta có thể dự đoán huyết áp tâm thu của 1 người ở độ tuổi bất kì.

Stochastic gradient descent

Giải:

- Giả sử hàm số có thể miêu tả mối liên hệ giữa tuổi và huyết áp tâm thu có dạng

$$b = x_0 + x_1 a$$

trong đó x_0, x_1 là các hệ số cần tìm.

- Ta mong muốn sự sai khác giá trị thực của huyết áp tâm thu và giá trị dự đoán là nhỏ nhất, tức là giá trị sau đây càng nhỏ càng tốt

$$\frac{1}{m}[b - (x_0 + x_1 a)]^2$$

- Với số liệu thống kê từ m người, ta có tổng sai số là

$$L(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m [b_i - (x_0 + x_1 a_i)]^2$$

trong đó $\mathbf{x} = (x_0, x_1)^T$ và ta gọi $L(\mathbf{x})$ là hàm mất mát (loss function).

Stochastic gradient descent

- Như vậy, ta cần tìm \mathbf{x} sao cho hàm mất mát $L(\mathbf{x})$ nhỏ nhất

$$\min_{\mathbf{x}} \frac{1}{m} \sum_{i=1}^m [b_i - (x_0 + x_1 a_i)]^2.$$

- Áp dụng phương pháp Stochastic gradient descent, ta đưa bài toán trên về dạng

$$\min_{\mathbf{x}} \frac{1}{m} \sum_{i=1}^m f_i(\mathbf{x})$$

trong đó

$$f_i(\mathbf{x}) = [b_i - (x_0 + x_1 a_i)]^2.$$

Stochastic gradient descent

Suy ra

$$\nabla f_i(\mathbf{x}) = \begin{pmatrix} -2[b_i - (x_0 + x_1 a_i)] \\ -2a_i[b_i - (x_0 + x_1 a_i)] \end{pmatrix}$$

Do đó, áp dụng công thức (1.2), ta có

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \nabla f_{i_t}(\mathbf{x}_t)$$

trong đó $i_t \in \{1, 2, \dots, m\}$ được chọn ngẫu nhiên trong mỗi lần cập nhật giá trị \mathbf{x}_{t+1} .

Stochastic gradient descent

Bài tập:

- a) Viết thuật toán Stochastic gradient descent cho bài toán trên
- b) Dùng Python để viết hàm thực hiện thuật toán Stochastic gradient descent ở câu a).
- c) Cho bảng số liệu sau

STT	Tuổi	HATT	STT	Tuổi	HATT
1	39	144	11	64	162
2	36	136	12	56	150
3	45	138	13	59	140
4	47	145	14	34	110
5	65	162	15	42	128
6	46	142	16	48	130
7	67	170	17	45	135
8	42	124	18	17	114
9	67	158	19	20	116
10	56	154	20	19	124

Áp dụng hàm đã viết ở câu b) để dự đoán huyết áp tâm thu của 1 người ở 1 độ tuổi bất kì.

Stochastic gradient descent

Giải:

a) Thuật toán:

Đầu vào: Độ tuổi a , huyết áp tâm thu b , learning rate η , giá trị ban đầu \mathbf{x}_t , bước lặp lớn nhất N , sai số ϵ

Đầu ra: Giá trị cực tiểu \mathbf{x}_{t+1} của hàm f .

- Bước 1: Tính vectơ gradient $\nabla f_{i_t}(\mathbf{x})$
- Bước 2: Đặt $i = 0$, $m = \text{len}(a)$.
- Bước 3: while $i \leq N$:
 - ▶ Bước 3.1: Chọn ngẫu nhiên $i_t \in \{1, 2, \dots, m\}$
 - ▶ Bước 3.2: Tính $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \nabla f_{i_t}(\mathbf{x}_t)$
 - ▶ Bước 3.3: Nếu $\|\nabla f(\mathbf{x}_{t+1})\|_2 < \epsilon$ thì
Xuất ra màn hình giá trị \mathbf{x}_{t+1}
Dừng lại
 - ▶ Bước 3.4: $\mathbf{x}_t = \mathbf{x}_{t+1}$, $i = i + 1$
- Bước 4: Xuất ra màn hình thông báo: thuật toán không thành công sau N bước lặp.