KHOA CNTT & TRUYỀN THÔNG
BM KHOA HỌC MÁY TÍNH

ÔN TẬP KIẾN THỨC TOÁN

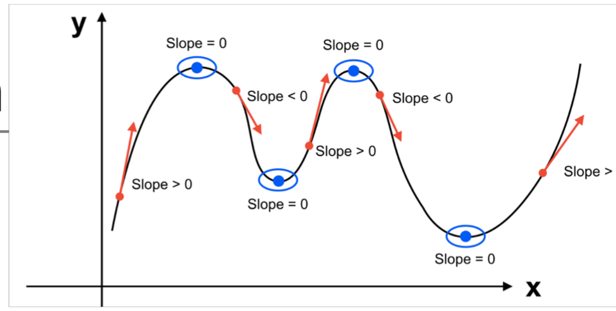
Nguyên lý máy học

Giáo viên giảng dạy:
TS. TRẦN NGUYỄN MINH THƯ
tnmthu@cit.ctu.edu.vn

NỘI DUNG

- Đạo hàm
- Đạo hàm riêng
- Phương pháp giảm gradient (Gradient descend)
- Phân tích hồi quy

Đạo hàm



Đạo hàm - derivative /dɪˈrɪvətɪv/

Đạo hàm của hàm số f(x) khảo sát sự biến thiên của hàm số f(x) theo biến x

Đạo hàm là công cụ giúp ta tìm được xem tại 1 điểm, đồ thị đang đi lên (Tăng) hay đi xuống (giảm) hay đạt cực đại, cực tiểu (không tăng- không giảm).

Đạo hàm

Đạo hàm của f(x) với x là biến số	Đạo hàm của f(u) với u là một hàm số
$(kx)' = k$	$(k \times u)' = k \times u'$
$(x^n)' = nx^{n-1}$	$(u^n)' = n \times u^{n-1} \times (u)'$
$\left(\frac{1}{x}\right)' = -\frac{1}{x^2}$	$\left(\frac{1}{u}\right)' = -\frac{(u)'}{u^2}$
$(\sqrt{x})' = \frac{1}{2\sqrt{x}}$	$(\sqrt{u})' = -\frac{(u)'}{2\sqrt{u}}$
$(\sin x)' = \cos x$	$(\sin\ u)' = \cos u \times (u)'$
$(\cos x)' = -\sin x$	$(\cos u)' = -\sin\ u \times (u)'$
$(\tan x)' = 1 + \tan^2 x = \frac{1}{\cos^2 x}$	$(\tan u)' = (1 + \tan^2 u) \times (u)' = \frac{(u)'}{\cos^2 u}$
$(\cot x)' = -(1 + \cot^2 x) = -\frac{1}{\sin^2 x}$	$(\cot u)' = -(1 + \cot^2 u) \times (u)' = -\frac{(u)'}{\sin^2 x}$
$(e^x)' = e^x$	$(e^u)' = e^u \times (u)'$
$(a^x)' = a^x \times \ln a$	$(a^u)' = a^u \times \ln a\ \times (u)'$
$(\ln x)' = (\ln\|x\|)' = \frac{1}{x}$	$(\ln u)' = (\ln\|u\|)' = \frac{(u)'}{u}$
$(\log_a x)' = (\log_a \|x\|)' = \frac{1}{x \times \ln a}$	$(\log_a u)' = (\log_a \|u\|)' = \frac{(u)}{u \times \ln a}$

Đạo hàm riêng

Đạo hàm riêng của một hàm số **đa biến** là đạo hàm theo một biến, các biến khác được xem như là hằng số.

Ký hiệu của đạo hàm riêng là

$$\frac{\partial f}{\partial x} \qquad \frac{d}{dx}$$

Ví dụ công thức sau: đạo hàm riêng của hàm số **J(θ)** theo **θ**$_j$

$$\frac{\partial}{\partial \theta_j} J(\theta)$$

https://vi.wikipedia.org/wiki/%C4%90%E1%BA%A1o_h%C3%A0m_ri%C3%AAng

Đạo hàm riêng: một số công thức cơ bản

$$\frac{d}{dx}(\alpha u) = \alpha \frac{du}{dx}$$

$$\frac{d}{dx}\Sigma u = \Sigma \frac{du}{dx}$$

$$\frac{d}{dx}u^n = nu^{n-1}\frac{du}{dx}$$

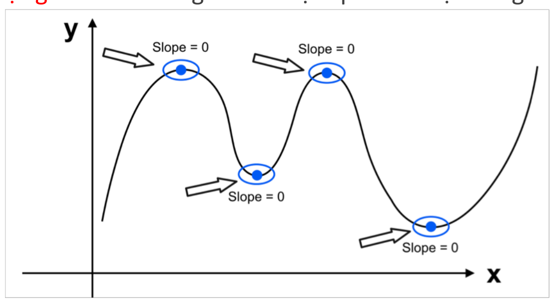
$$\frac{d}{dx}f(g(x)) = f'\big(g(x)\big)g'(x)$$

Giảm Gradient

Để tìm cực đại, cực tiểu, phương pháp thông thường trong toán học là ta giải phương trình

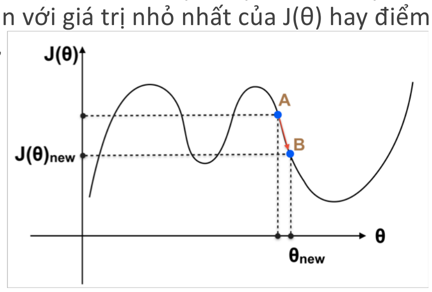
$$J'(\theta) = 0$$

Tuy nhiên phương trình không phải lúc nào cũng dễ giải và chi phí để giải phương trình khi số lượng biến lớn cũng khiến hiệu quả của hệ thống thấp.



Giảm Gradient

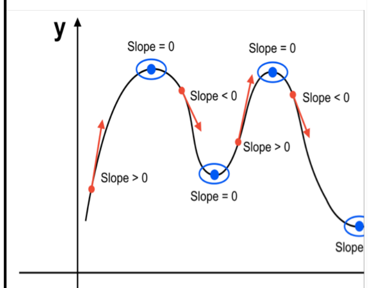
Vấn để được đặt ra là nếu không giải phương trình, với giá trị θ hiện tại (chẳng hạn ban đầu tất cả tham số θ = 0) làm sao để thay đổi θ để cho J(θ) nhỏ hơn (chưa phải nhỏ nhất), nếu như lặp lại quá trình này nhiều lần thì ta sẽ chọn được θ sao cho J(θ) gần với giá trị nhỏ nhất của J(θ) hay điểm cực tiểu của hàm số. Giả sử với θ hiện tại,

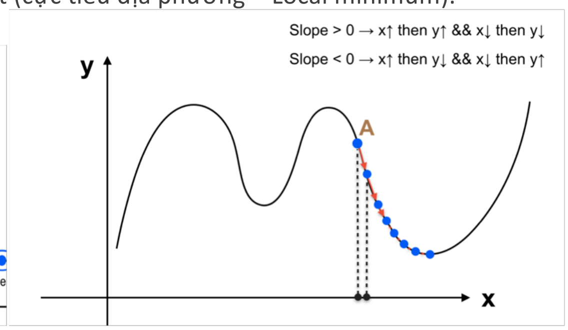


Với θ hiện tại, J(θ) đang ở điểm A, rõ ràng ở vị trí A đồ thị trên đang có xu hướng giảm (đạo hàm hay slope tại điểm A < 0, nếu tăng θ thì J(θ) sẽ giảm). Ta tăng θ để A tiến đến điểm B như hình

Giảm Gradient

Khi đó J(θ) giảm xuống (vì slope tại A 0 thì khi tăng θ thì J(θ) sẽ tăng, lúc này muốn giảm J(θ) ta phải giảm θ. Vậy với giá trị θ hiện tại, chúng ta đã biết tăng hay giảm θ để giá trị của J(θ) giảm xuống. Thử tưởng tượng rằng chúng ta làm việc này một cơ số lần, khi đó J(θ) sẽ giảm dần xuống giá trị cực tiểu gần nhất (cực tiểu địa phương – Local minimum).





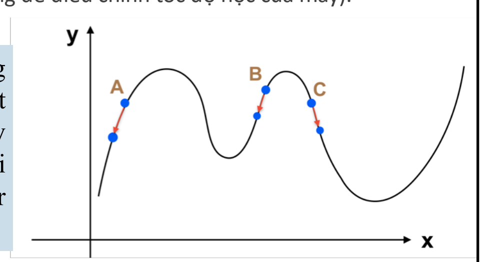
Giảm Gradient

Giải thuật giảm Gradient được mô tả như sau:

+ Khởi tạo θ bất kì (thường tất cả tham số θ được gán = 0).
+ Liên tiếp thay đổi **θ := θ – α * J'(θ)** đồng thời (tất cả các tham số θ phải được thay đổi cùng 1 lúc)
+ Dừng lại khi J(θ) có thay đổi không đáng kể.

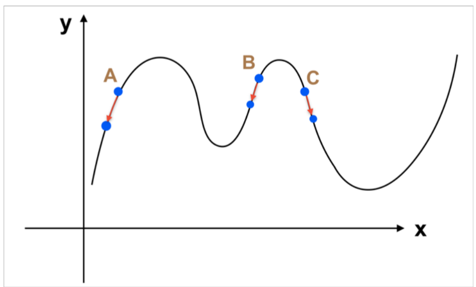
Trong đó α được gọi là Learning rate (dùng để điều chỉnh tốc độ học của máy).

Gradient descent (giảm gradient) là phương pháp dùng để tìm cực tiểu cục bộ của một hàm. Ý tưởng chính của phương pháp này là từ vị trí hiện tại, thực hiện một bước đi nhỏ, ngược lại với hướng của vector gradient tại vị trí đó.



Giảm Gradient

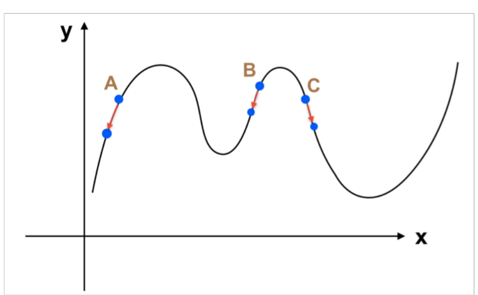
Giải thuật được mô tả như sau:
+ Khởi tạo θ bất kì (thường tất cả tham số θ được gán = 0).
+ Liên tiếp thay đổi θ := θ – α * J'(θ) đồng thời (tất cả các tham số θ phải được thay đổi cùng 1 lúc)
+ Dừng lại khi J(θ) có thay đổi không đáng kể.
Trong đó α được gọi là Learning rate (dùng để điều chỉnh tốc độ học của máy). Trước khi đi vào giải thích về tham số này, chúng ta nói một chút về nhược điểm của giải

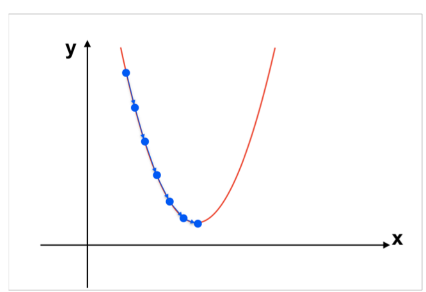


+ Khi ta đang ở điểm A, có thể giải thuật này sẽ đi vào vòng lặp vô hạn do hàm số đi xuống mãi khi giảm θ. Tuy nhiên trong trường hợp này việc giải phương trình cũng không mang lại kết quả do bản thân hàm số J(θ) không có giá trị nhỏ nhất (Global minimum). Cách giải quyết là ta phải xây dựng hàm J(θ) có giá trị nhỏ nhất.

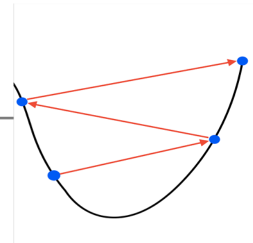
Giảm Gradient

Khi ta đang ở điểm B và điểm C, áp dụng giải thuật trên sẽ đưa ta đến 2 điểm cực tiểu khác nhau. Rõ ràng tại C, ta sẽ giảm được J(θ) nhiều hơn, chính vì thế để đảm bảo J(θ) đạt được giá trị nhỏ nhất, ta phải cố gắng xây dựng J(θ) có dạng tương tự như parabolla. Khi đó J(θ) sẽ chỉ có một giá trị nhỏ nhất, và giải thuật chắc chắn đưa J(θ) xuống đáy của đồ thị.





Giảm Gradient – tốc độ học



Tốc độ học - Learning rate α.

Nếu như ta loại bỏ tham số α thì chẳng qua là để α = 1

giải thuật trở thành $\theta := \theta - J'(\theta)$

Việc xác định được các tham số θ nhanh hay chậm phụ thuộc vào tham số này, tuy nhiên nếu như lựa chọn không phù hợp có thể gây ra hệ luỵ nhất định. Cụ thể như sau:

- α quá nhỏ: α nhỏ sẽ khiến cho các bước đi xuống dốc của J(θ) cũng nhỏ => việc học mất nhiều thời gian hơn
- α quá lớn: α lớn sẽ khiến cho các bước đi trở nên quá lớn, có thể khiến J(θ) lớn hơn giá trị hiện tại.