

Đạo hàm và gradient

☑ Trắc nghiệm 7 ⌵ PDF bài

Một model đoán điểm bằng $\hat{y} = wx$ với một trọng số w . Đặt $w = 0$ thì đoán sai nhiều, loss lớn. Nên tăng hay giảm w , và đổi bao nhiêu? Với một trọng số thì có thể thử từng giá trị, nhưng mạng neural có hàng triệu trọng số. Đạo hàm cho biết đổi w một chút thì loss đổi bao nhiêu và theo hướng nào. Bài này đi từ đạo hàm tới gradient và gradient descent, thuật toán dùng để huấn luyện gần như mọi model trong khóa học.

MỤC TIÊU

- Hiểu đạo hàm là độ dốc; tính xấp xỉ bằng sai phân.
- Tính đạo hàm của đa thức, e^x , $\log x$ và hàm hợp.
- Tính gradient của hàm nhiều biến.
- Chạy gradient descent bằng tay và bằng code; chọn learning rate.

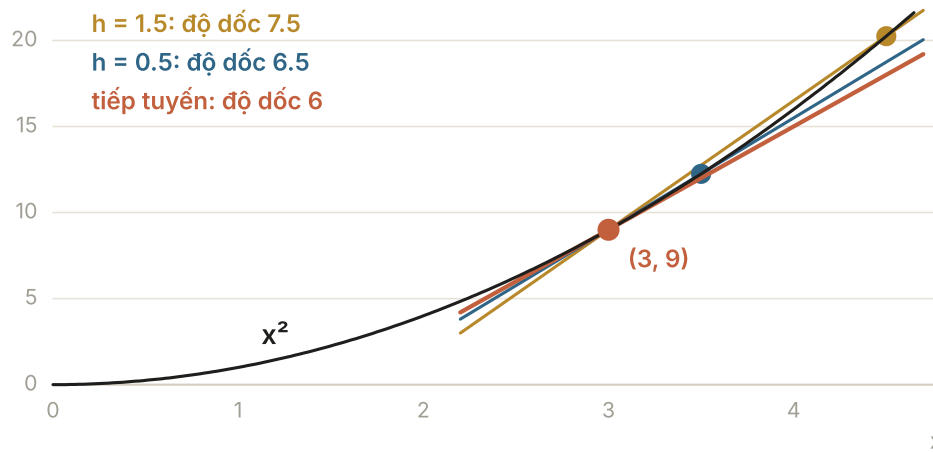
Độ dốc tại một điểm

Ở bài 0.4, hàm bậc nhất có độ dốc không đổi: x tăng 1 thì y tăng w . Hàm $f(x) = x^2$ thì khác: gần $x = 0$ đồ thị gần như nằm ngang, càng xa thì càng dốc. Ta cần nói “độ dốc **tại** $x = 3$ ”.

Cách làm: lấy một điểm gần đó, $x = 3 + h$, và chia độ tăng của f cho độ tăng của x :

$$\frac{f(3 + h) - f(3)}{h}.$$

Đây là độ dốc của đường thẳng nối hai điểm trên đồ thị. Cho h nhỏ dần, đường ấy xoay dần về **tiếp tuyến**, đường thẳng chỉ chạm đồ thị tại đúng điểm $x = 3$. Độ dốc của tiếp tuyến là **đạo hàm** của f tại 3, ký hiệu $f'(3)$.



h càng nhỏ, đường nối hai điểm càng gần tiếp tuyến; độ dốc tiến về 6.

Python

```
import numpy as np

def f(x):
    return x ** 2

for h in [1, 0.1, 0.01, 0.001]:
    print(h, round((f(3 + h) - f(3)) / h, 6))
```

KẾT QUẢ

```
1 7.0
0.1 6.1
0.01 6.01
0.001 6.001
```

Với h nhỏ dần, tỉ số tiến về 6. Viết tay: $\frac{(3+h)^2-9}{h} = \frac{6h+h^2}{h} = 6 + h$, và $h \rightarrow 0$ cho đúng 6.

Đạo hàm như một lời hứa nhỏ: **tăng x thêm một lượng rất nhỏ Δ thì f tăng thêm khoảng $f'(x) \cdot \Delta$** . Đạo hàm dương thì hàm đang đi lên, âm thì đang đi xuống, bằng 0 thì đang nằm ngang, như ở đáy một cái bát.

HỎI NHANH

$f'(2) = -3$. Tăng x từ 2 lên 2.01 thì f thay đổi khoảng bao nhiêu?

- A Tăng khoảng 0.03
- B Giảm khoảng 0.03
- C Giảm khoảng 3
- D Không đổi, vì 0.01 quá nhỏ

Các quy tắc tính đạo hàm

Không ai tính giới hạn mỗi lần. Vài quy tắc đủ cho phần lớn học máy:

Hàm	Đạo hàm
hằng số c	0
x^n	nx^{n-1}
e^x	e^x
$\log x$	$1/x$
$c \cdot f(x)$	$c \cdot f'(x)$
$f(x) + g(x)$	$f'(x) + g'(x)$

Hàng x^n cho $(x^2)' = 2x$, khớp con số 6 tại $x = 3$. Hàng e^x là lý do bài 0.4 chọn cơ số e : đạo hàm của nó là chính nó. Hai hàng cuối nói đạo hàm đi qua phép nhân với hằng số và phép cộng: $(3x^2 - 4x + 1)' = 6x - 4$.

Kiểm bằng số: so công thức với sai phân tại $x = 2$.

Python

```
def numeric(g, x, h=1e-6):
    return (g(x + h) - g(x - h)) / (2 * h)

x = 2.0
tests = [("x^3", lambda t: t ** 3, 3 * x ** 2),
         ("e^x", np.exp, np.exp(x)),
         ("log x", np.log, 1 / x),
         ("3x^2 - 4x + 1", lambda t: 3 * t ** 2 - 4 * t + 1, 6 * x - 4)]
for name, g, exact in tests:
    print(f"{name:14s} công thức {exact:.6f} sai phân {numeric(g, x):.6f}")
```

KẾT QUẢ

x^3	công thức 12.000000	sai phân 12.000000
e^x	công thức 7.389056	sai phân 7.389056
$\log x$	công thức 0.500000	sai phân 0.500000
$3x^2 - 4x + 1$	công thức 8.000000	sai phân 8.000000

Hàm `numeric` lấy hai điểm hai bên, $x + h$ và $x - h$, gọi là **sai phân trung tâm**; nó chính xác hơn cách lấy một bên ở mục trước. Mục cuối bài dùng lại hàm này để kiểm gradient.

HỎI NHANH

$g(x) = 4x^3 - 2x$. $g'(1)$ bằng bao nhiêu?

A 2

B 12

C 14

D 10

Quy tắc chuỗi

Loss của model hiếm khi là một hàm đơn giản của trọng số. Nó là một **chuỗi** các bước: trọng số cho ra dự đoán, dự đoán cho ra sai số, sai số cho ra loss. Với hàm hợp $f(g(x))$, quy tắc chuỗi nói

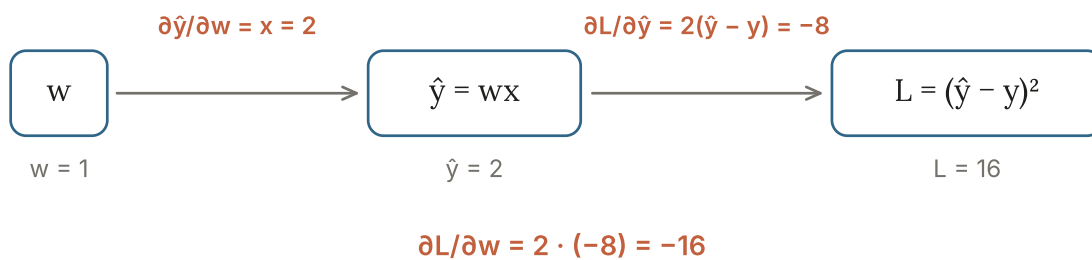
$$\frac{d}{dx}f(g(x)) = f'(g(x)) \cdot g'(x).$$

Đọc bằng lời hứa ở trên: x nhích Δ thì g nhích khoảng $g'(x)\Delta$, và g nhích chừng ấy thì f nhích khoảng $f'(g)$ nhân lượng đó. Tốc độ đổi qua nhiều bước là **tích** các tốc độ đổi của từng bước.

Ví dụ $h(x) = (2x + 1)^2$. Bên trong là $g = 2x + 1$ với $g' = 2$; bên ngoài là $f = g^2$ với $f' = 2g$. Nên $h'(x) = 2(2x + 1) \cdot 2 = 4(2x + 1)$, tại $x = 1$ bằng 12.

Áp vào model $\hat{y} = wx$ với loss $L = (\hat{y} - y)^2$. Coi w là biến, x và y là số đã biết. Chuỗi gồm hai bước: $w \rightarrow \hat{y}$ có đạo hàm x , và $\hat{y} \rightarrow L$ có đạo hàm $2(\hat{y} - y)$. Nhân lại:

$$\frac{\partial L}{\partial w} = 2(\hat{y} - y) x.$$



Đạo hàm của loss theo w bằng tích đạo hàm của từng bước.

Mạng neural chỉ là một chuỗi dài hơn. Thuật toán backpropagation ở chương 3 là quy tắc chuỗi áp dụng có hệ thống, từ loss đi ngược về từng trọng số.

Một ví dụ quan trọng là sigmoid $\sigma(z) = (1 + e^{-z})^{-1}$. Áp quy tắc chuỗi hai lần rồi rút gọn, ta được một công thức rất gọn:

$$\sigma'(z) = \sigma(z)(1 - \sigma(z)).$$

Tại $z = 0$, $\sigma = 0.5$ nên $\sigma' = 0.25$, lớn nhất có thể. Ở hai đầu chữ S, σ' gần 0. Một mạng nhiều lớp sigmoid nhân nhiều thừa số không quá 0.25 với nhau, nên gradient ở các lớp đầu nhỏ dần. Chương 3 gọi đây là vanishing gradient.

```
def sigmoid(z):
    return 1 / (1 + np.exp(-z))

for z in [0.0, 2.0, 5.0]:
    s = sigmoid(z)
    print(z, round(s * (1 - s), 4), round(numeric(sigmoid, z), 4))

w, x, y = 1.0, 2.0, 6.0
loss = lambda w: (w * x - y) ** 2
print(2 * (w * x - y) * x, round(numeric(loss, w), 4))
```

KẾT QUẢ

```
0.0 0.25 0.25
2.0 0.105 0.105
5.0 0.0066 0.0066
-16.0 -16.0
```

Công thức và sai phân khớp nhau ở cả ba điểm của sigmoid. Dòng cuối: với $w = 1$, model đoán 2 trong khi đáp án là 6; đạo hàm -16 âm, nghĩa là **tăng** w thì loss giảm. Đúng như mong đợi, vì model đang đoán thấp.

HỎI NHANH

$L = (wx - y)^2$ với $x = 3$, $y = 3$ và $w = 2$. $\partial L / \partial w$ bằng bao nhiêu, và nên đổi w theo hướng nào?

A 6; tăng w

B 9; giảm w

C 18; giảm w

D -18 ; tăng w

Đạo hàm riêng và gradient

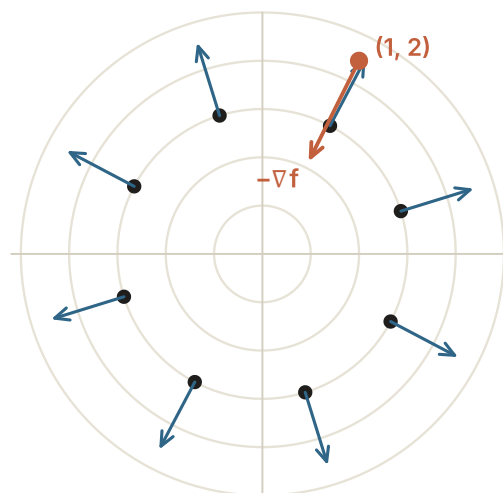
Loss thường phụ thuộc nhiều tham số cùng lúc. Với hàm hai biến $f(x, y)$, **đạo hàm riêng** theo x , viết $\partial f / \partial x$, là đạo hàm khi coi y như hằng số. Ví dụ $f(x, y) = x^2y + 3x$:

$$\frac{\partial f}{\partial x} = 2xy + 3, \quad \frac{\partial f}{\partial y} = x^2.$$

Xếp mọi đạo hàm riêng thành một vector, ta được **gradient**:

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right).$$

Gradient chỉ **hướng làm f tăng nhanh nhất**, và độ dài của nó cho biết dốc cỡ nào. Hướng ngược lại, $-\nabla f$, làm f giảm nhanh nhất. Hình dưới là bản đồ đường đồng mức của cái bát $f(x, y) = x^2 + y^2$; mỗi vòng là một mức cao. Gradient tại mỗi điểm vuông góc với đường đồng mức và chỉ ra ngoài.



$\nabla f = (2x, 2y)$
chỉ ra xa đáy

Gradient của $x^2 + y^2$ chỉ ra xa đáy; $-\nabla f$ chỉ về đáy.

Python

```
def bowl(v):  
    return v[0] ** 2 + v[1] ** 2  
  
def grad_numeric(g, v, h=1e-6):  
    out = np.zeros_like(v)
```

```
for i in range(len(v)):
    e = np.zeros_like(v)
    e[i] = h
    out[i] = (g(v + e) - g(v - e)) / (2 * h)
return out
```

```
v = np.array([1.0, 2.0])
print(2 * v, np.round(grad_numeric(bowl, v), 6))
```

KẾT QUẢ

[2. 4.] [2. 4.]

grad_numeric làm sai phân trung tâm theo từng tọa độ, giữ các tọa độ khác cố định: đúng định nghĩa đạo hàm riêng. Tại (1, 2), gradient là (2, 4), chỉ ra xa gốc.

HỎI NHANH

$f(x, y) = 3x + y^2$. Gradient tại (5, 1) là gì?

A (3, 1)

B (3, 2)

C (15, 1)

D 5

Gradient descent

Đã biết hướng làm loss giảm nhanh nhất là $-\nabla L$. Gradient descent đi một bước nhỏ theo hướng ấy, rồi tính lại gradient, rồi đi tiếp:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla L(\mathbf{w}).$$

η là **learning rate**, quyết định mỗi bước dài bao nhiêu. Hình ảnh quen thuộc: đi xuống núi trong sương mù, chỉ thấy độ dốc dưới chân, nên mỗi lần bước một bước theo hướng

dốc xuống nhất.

Thử với model một trọng số ở trên, $x = 2$, $y = 6$, đáp án tốt nhất là $w = 3$. Bắt đầu từ $w = 0$, so hai learning rate.

Python

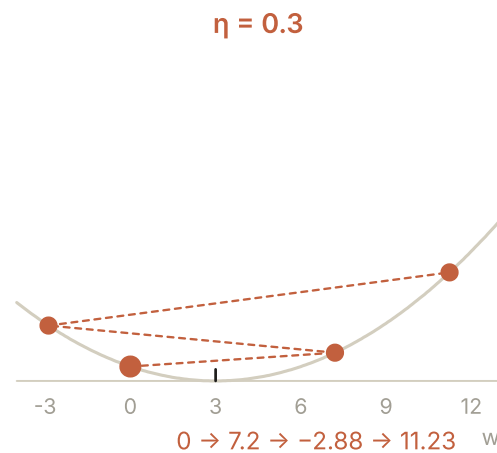
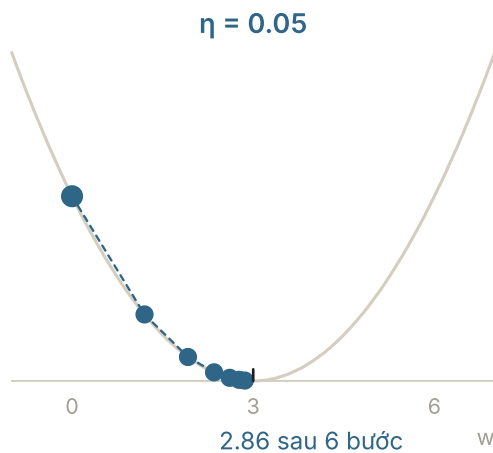
```
def descend(eta, steps, w=0.0):  
    path = []  
    for _ in range(steps):  
        grad = 2 * (w * x - y) * x  
        w = w - eta * grad  
        path.append(round(w, 4))  
    return path  
  
print(descend(0.05, 6))  
print(descend(0.05, 30)[-1])  
print(descend(0.3, 6))
```

KẾT QUẢ

[1.2, 1.92, 2.352, 2.6112, 2.7667, 2.86]

3.0

[7.2, -2.88, 11.232, -8.5248, 19.1347, -19.5886]



$\eta = 0.05$ đi dần xuống đáy; $\eta = 0.3$ nhảy qua đáy, văng xa dần.

Với $\eta = 0.05$, w tiến dần về 3, bước sau ngắn hơn bước trước vì càng gần đáy thì gradient càng nhỏ. Sau 30 bước đã bằng 3 tới bốn chữ số thập phân. Với $\eta = 0.3$, bước đầu nhảy lên 7.2, vượt qua đáy rất xa; mỗi bước sau lại vượt sang phía bên kia, xa hơn bước trước,

và w lớn dần không giới hạn. Ta nói gradient descent **phân kỳ**. Khi loss tăng vọt hay thành nan trong lúc huấn luyện, nghi phạm đầu tiên là learning rate quá lớn.

Với nhiều mẫu, loss là trung bình loss của từng mẫu, nên gradient cũng là trung bình gradient của từng mẫu. Lấy hai mẫu $(x, y) = (1, 2)$ và $(2, 5)$, bắt đầu từ $w = 0$ với $\eta = 0.1$. Hai dự đoán đều là 0, sai số -2 và -5 , loss $(4 + 25)/2 = 14.5$. Gradient của từng mẫu là $2 \cdot \text{sai số} \cdot x$: -4 và -20 , trung bình -12 . Bước cập nhật cho $w = 0 - 0.1 \cdot (-12) = 1.2$. Sai số mới là -0.8 và -2.6 , loss còn $(0.64 + 6.76)/2 = 3.7$.

Python

```
xs = np.array([1.0, 2.0])
ys = np.array([2.0, 5.0])
w = 0.0
for step in range(3):
    err = w * xs - ys
    grad = np.mean(2 * err * xs)
    print(step, round(w, 4), round(np.mean(err ** 2), 4), round(grad, 4))
    w -= 0.1 * grad
```

KẾT QUẢ

```
0 0.0 14.5 -12.0
1 1.2 3.7 -6.0
2 1.8 1.0 -3.0
```

Dòng 0 và dòng 1 khớp phép tính tay. Đáp án tốt nhất là $w = 12/5 = 2.4$, nơi gradient trung bình bằng 0; vài bước nữa w sẽ tới đó.

HỎI NHANH

Huấn luyện một model bằng gradient descent, loss sau mỗi lượt chạy hết dữ liệu (gọi là một epoch) lần lượt là 2.1, 5.8, 31.4, 260.0, rồi nan. Nên thử gì trước?

- A Giảm learning rate
- B Tăng learning rate cho nhanh tới đáy
- C Huấn luyện thêm nhiều epoch
- D Bỏ bớt dữ liệu train

Kiểm gradient bằng sai phân

Tự viết công thức gradient thì dễ sai một hệ số 2 hay một dấu trừ, mà code vẫn chạy. Cách kiểm: so gradient từ công thức với gradient tính bằng sai phân trên một ví dụ nhỏ. Nếu sai số tương đối cỡ 10^{-6} trở xuống thì công thức gần như chắc đúng.

Model tuyến tính ba trọng số, loss MSE trên năm mẫu: $L(\mathbf{w}) = \frac{1}{n} \|X\mathbf{w} - \mathbf{y}\|^2$. Quy tắc chuỗi cho gradient

$$\nabla L = \frac{2}{n} X^\top (X\mathbf{w} - \mathbf{y}).$$

Đọc bằng shape của bài 0.5: $X^\top$ là (3, 5), phần dư $X\mathbf{w} - \mathbf{y}$ là (5,), tích là (3,), đúng một số cho mỗi trọng số.

Python

```
rng = np.random.default_rng(0)
X = rng.normal(size=(5, 3))
y_vec = rng.normal(size=5)
w_vec = rng.normal(size=3)

def mse(w):
    return np.mean((X @ w - y_vec) ** 2)
```

```

exact = 2 / len(y_vec) * X.T @ (X @ w_vec - y_vec)
approx = grad_numeric(mse, w_vec)
wrong = 1 / len(y_vec) * X.T @ (X @ w_vec - y_vec)      # quên số 2
rel = lambda a, b: np.linalg.norm(a - b) / np.linalg.norm(a + b)
print(np.round(exact, 6))
print(np.round(approx, 6))
print(f"đúng: {rel(exact, approx):.1e}    quên số 2: {rel(wrong, approx):.1e}")

```

KẾT QUẢ

```

[ 1.753666  1.112768 -0.48581 ]
[ 1.753666  1.112768 -0.48581 ]
đúng: 3.1e-11    quên số 2: 3.3e-01

```

Công thức đúng cho sai số tương đối rất nhỏ; bản quên số 2 cho sai số cỡ 0.3, lộ ngay. Đây cũng là cách PyTorch kiểm các phép tính tự viết (`torch.autograd.gradcheck`).

HỎI NHANH

Kiểm gradient tự viết bằng sai phân, sai số tương đối ra 0.5. Nhiều khả năng do đâu?

- A Bình thường, vì sai phân luôn chỉ là xấp xỉ
- B Phải giảm h xuống 10^{-15} cho chính xác hơn
- C Công thức gradient tự viết nhiều khả năng sai
- D Learning rate đang đặt quá lớn

Khi đi thi

- Tự viết gradient thì kiểm bằng sai phân trung tâm trên ví dụ nhỏ, $h \approx 10^{-6}$.
- Gradient phải có đúng shape của tham số.
- Loss tăng vọt hay thành nan thì giảm learning rate trước.

Tóm tắt

- Đạo hàm $f'(x)$ là giới hạn của $\frac{f(x+h)-f(x)}{h}$, độ dốc tiếp tuyến; x nhích Δ thì f nhích khoảng $f'(x)\Delta$.
- $(x^n)' = nx^{n-1}$, $(e^x)' = e^x$, $(\log x)' = 1/x$; đạo hàm đi qua phép cộng và nhân hằng số.
- Quy tắc chuỗi: $(f(g(x)))' = f'(g(x))g'(x)$; $\sigma' = \sigma(1 - \sigma) \leq 0.25$.
- Gradient là vector các đạo hàm riêng, chỉ hướng tăng nhanh nhất; $-\nabla f$ chỉ hướng giảm nhanh nhất.
- Gradient descent: $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla L$; η quá lớn thì phân kỳ.
- MSE tuyến tính có $\nabla L = \frac{2}{n} X^\top (X\mathbf{w} - \mathbf{y})$; kiểm mọi gradient tự viết bằng sai phân trung tâm.

Kiểm tra cuối bài

Mười câu dưới đây bao quát cả bài. Làm hết rồi xem điểm; câu nào sai thì đọc lại mục tương ứng.

CÂU 1

Đạo hàm của $f(x) = 5x + 3$ là gì?

A 8

B 5

C 3

D $5x$

ĐỀ THI NOAI 2026 Singapore · Vòng sơ loại · Câu 38

CÂU 2

Quy tắc chuỗi nói $\frac{d}{dx} f(g(x))$ bằng gì?

A $f'(x) \cdot g'(x)$

B $f'(x) + g'(x)$

C $f(x) \cdot g'(x)$

D $f'(g(x)) \cdot g'(x)$

ĐỀ THI NOAI 2026 Singapore · Vòng sơ loại · Câu 39

CÂU 3

Gradient của $f(x, y) = x^2 + y^2$ là gì?

A $(2x, 2y)$

B (x, y)

C (x^2, y^2)

D $2x + 2y$

ĐỀ THI NOAI 2026 Singapore · Vòng sơ loại · Câu 41

CÂU 4

Đạo hàm của e^x là gì?

A xe^{x-1}

B e

C e^x

D 1

ĐỀ THI NOAI 2026 Singapore · Vòng sơ loại · Câu 42

CÂU 5

Trong gradient descent, vector âm của gradient, $-\nabla L$, chỉ về đâu?

A Hướng làm loss giảm nhanh nhất

B Hướng làm loss tăng nhanh nhất

C Thẳng tới điểm cực tiểu của loss

D Một điểm yên ngựa của loss

ĐỀ THI NOAI 2026 Singapore · Vòng sơ loại · Câu 44

CÂU 6

Đạo hàm của sigmoid $\sigma(x) = 1/(1 + e^{-x})$ theo x là gì?

A $\sigma(x)$

B e^{-x}

C $-\sigma(x)$

D $\sigma(x)(1 - \sigma(x))$

ĐỀ THI NOAI 2026 Singapore · Vòng sơ loại · Câu 45

CÂU 7

Với $L = (\hat{y} - y)^2$ và $\hat{y} = wx$, $\partial L / \partial w$ bằng gì?

A $2(\hat{y} - y)$

B $(\hat{y} - y)x$

C $2(\hat{y} - y)x$

D $2w$

ĐỀ THI NOAI 2026 Singapore · Vòng sơ loại · Câu 46

CÂU 8

Gradient của $f(\mathbf{x}) = \|\mathbf{x}\|^2$, bình phương chuẩn L2, là gì?

A $\mathbf{x}$

B $\|\mathbf{x}\|$

C $\mathbf{x}^2$

D $2\mathbf{x}$

ĐỀ THI NOAI 2026 Singapore · Vòng sơ loại · Câu 53

CÂU 9

Loss $L(w) = (w - 4)^2$, đang ở $w = 0$, learning rate 0.25. Sau một bước gradient descent, w bằng bao nhiêu?

A -2

B 1

C 2

D 4

CÂU 10

Một học sinh kiểm gradient bằng sai phân một bên $\frac{f(x+h)-f(x)}{h}$ và thấy sai số nhỏ dần khi giảm h từ 10^{-2} xuống 10^{-6} , nhưng tăng lên khi $h = 10^{-12}$. Vì sao?

- A h quá nhỏ thì phép trừ mất chữ số do làm tròn
- B Công thức gradient sai, nên chỉ đúng ở vài giá trị h
- C Sai phân chỉ đúng khi h là số nguyên
- D Hàm không có đạo hàm tại điểm đang kiểm

Đọc thêm

- 3Blue1Brown — [Essence of calculus](#), tập 1–4. Đạo hàm như độ dốc và quy tắc chuỗi, bằng hình động.
- 3Blue1Brown — [Gradient descent, how neural networks learn](#). Gradient descent trên một mạng neural thật, chuẩn bị cho chương 3.
- Khan Academy — [Differential calculus](#). Bài giảng và bài tập luyện quy tắc đạo hàm.
- Deisenroth, Faisal, Ong — [Mathematics for Machine Learning](#), chương 5. Gradient của hàm vector và ma trận, khi cần đi xa hơn bài này.
- Stanford CS231n — [Optimization: stochastic gradient descent](#). Kiểm gradient bằng sai phân và chọn learning rate trong thực tế.