

Nén một chiều, nâng một chiều

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$2 \times 3: \mathbb{R}^3 \rightarrow \mathbb{R}^2$$

$$E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$$

$$3 \times 2: \mathbb{R}^2 \rightarrow \mathbb{R}^3$$

$$P \cdot (1, 2, 5) = (1, 2)$$

$$P \cdot (1, 2, 99) = (1, 2) \leftarrow \text{GIỐNG HẾT}$$

$$E \cdot (3, 2) = (3, 2, 5)$$

$$P \cdot E = I_2$$

$$E \cdot P = [[1, 0, 0], [0, 1, 0], [1, 1, 0]] \neq I_3$$

Toàn bộ bài trong hai câu

- Hai đầu vào khác nhau, cùng một đầu ra: bạn vừa NHÌN THẤY mất thông tin.
- Nâng rồi nén thì hồi được; nén rồi nâng thì không.

Phần còn lại: vì sao, và vì sao nó CHÍNH LÀ embedding.

☆ Embedding, nn.Embedding, $Q = X \cdot W_Q$, LoRA:
tất cả chỉ là một ma trận $m \times n$ với $m \neq n$.

$m \times n$: vào n chiều, ra m chiều

Số CỘT = chiều VÀO. Số HÀNG = chiều RA.

Đọc " m nhân n ", nhưng chiều đi là $n \rightarrow m$: ngược thứ tự chữ. Bấy số 1.

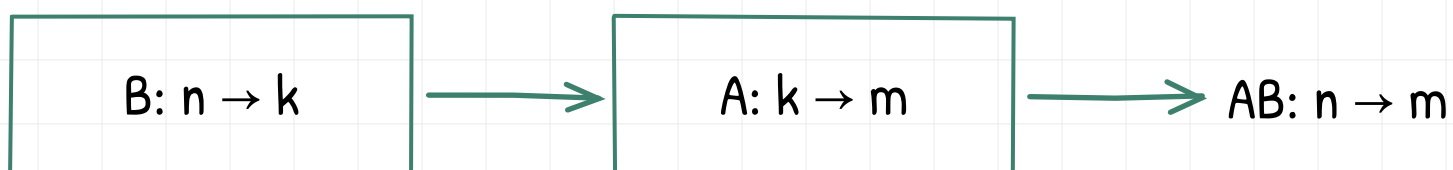
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \longrightarrow \begin{matrix} 2 \text{ hàng: mỗi cột là} \\ \text{một điểm của } \mathbb{R}^2 \end{matrix}$$

3 cột: 3 vector cơ sở đầu vào

Mỗi cột vẫn là "nơi một vector cơ sở đáp xuống" (bài 2).

Chỉ khác: chỗ đáp xuống nằm ở KHÔNG GIAN KHÁC.

Ma trận = hàm có chữ ký $(n) \Rightarrow (m)$



$A (m \times k) \cdot B (k \times n) \checkmark$ ra $(m \times n)$: "chiều trong" khớp nhau.

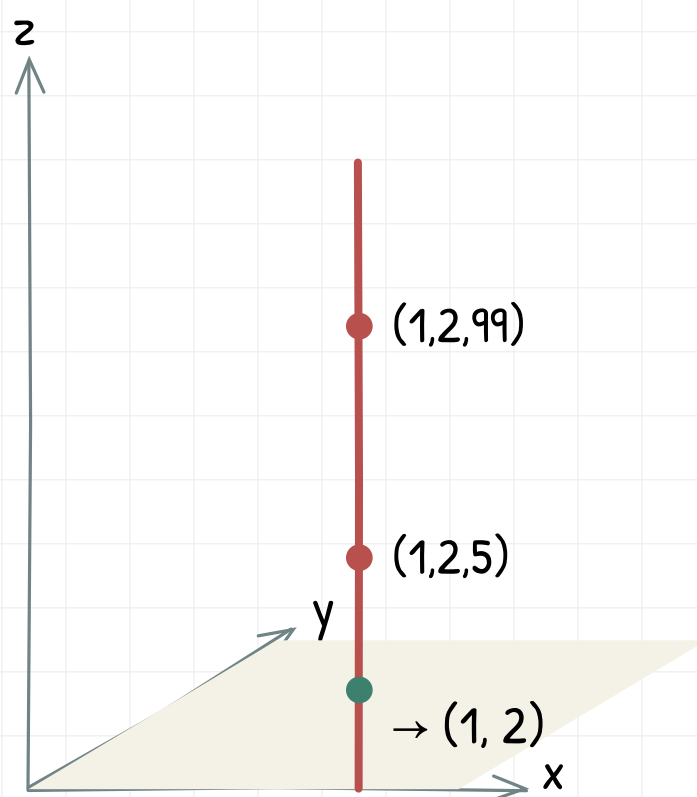
$A (2 \times 3) \cdot B (2 \times 3) \times$ lỗi chiều.

Không phải luật tùy tiện: đầu ra hàm này phải đúng kiểu đầu vào hàm kia.

Y như compose hai function trong TypeScript.

Ma trận vuông $(n) \Rightarrow (n)$ mới ghép chuỗi thoải mái.

Nén ($m < n$): mất thông tin là BẮT BUỘC



Cột 3 của $P = (0, 0)$:
trục z bị nghiền về gốc.

Cả đường đứng $(1, 2, t)$
dồn về MỘT điểm $(1, 2)$.

$$(1, 2, 99) - (1, 2, 5) = (0, 0, 94) \neq 0, \quad P \cdot (0, 0, 94) = (0, 0)$$

$\Rightarrow \text{null}(P) = \text{trục } z \text{ (1 chiều)}, \text{ y như bài 3 §6.}$

hạng + số chiều null = số chiều ĐẦU VÀO

$$2 + 1 = 3. \quad \text{hạng} \leq \min(m, n) \text{ luôn luôn.}$$

$m < n \Rightarrow \text{hạng} \leq m < n \Rightarrow \text{null có ít nhất } n - m \text{ chiều.}$

Không phải ma trận chọn tồi: nhét n chiều vào m ô (pigeonhole).

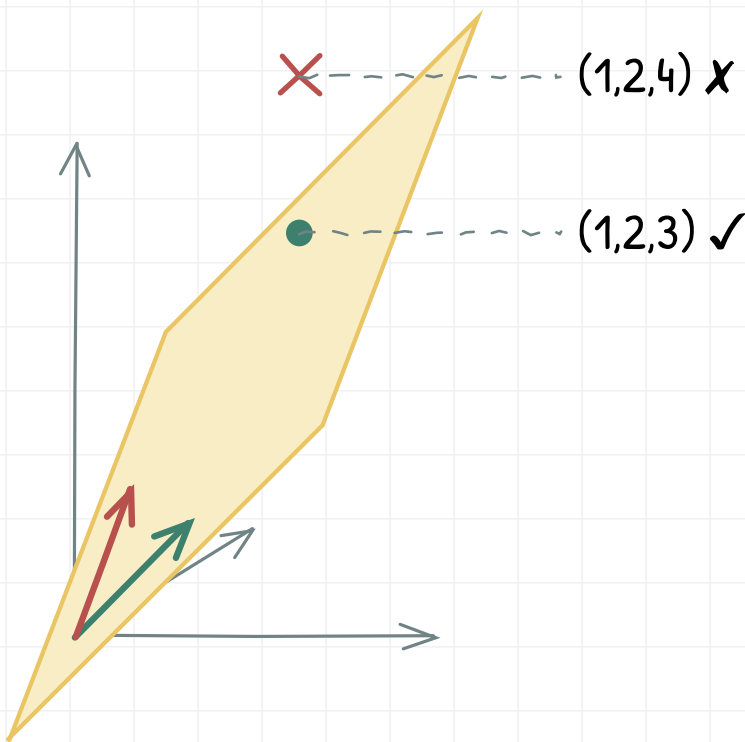
Như md5: đầu ra 128 bit, đầu vào vô hạn $\Rightarrow$ không tồn tại md5.undo().

Bỏ qua được: tìm cơ sở null bằng tay cho ma trận lớn (scipy làm hộ).

Nâng (m > n): không mất, không phủ hết

$E \cdot (1, 0) = (1, 0, 1)$
 $E \cdot (0, 1) = (0, 1, 1)$
 $E \cdot (3, 2) = 3 \cdot (1, 0, 1) + 2 \cdot (0, 1, 1) = (3, 2, 5)$
 $E \cdot (1, 2) = (1, 2, 3)$

$E \cdot (a, b) = (a, b, a + b) \Rightarrow z = x + y$



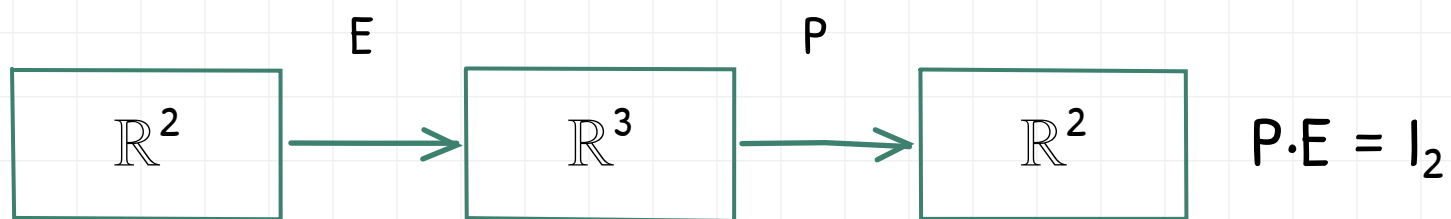
Ảnh của E: mặt phẳng
 $x + y - z = 0$
(pháp tuyến $(1, 1, -1)$).

Một tờ giấy 2 chiều
lơ lửng trong $\mathbb{R}^3$.

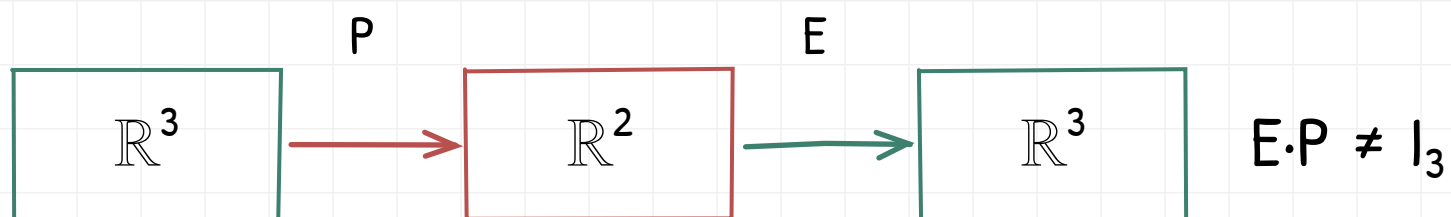
điểm	$x + y - z$	trong ảnh?
$(1, 2, 3)$	0	có = $E \cdot (1, 2)$
$(1, 2, 4)$	-1	không
$(0, 0, 1)$	-1	không

$\text{hạng}(E) = 2 = \min(3, 2)$: kịch trần. Nâng chiều không TẠO thông tin, chỉ đặt nó vào chỗ rộng hơn. Muốn lấp đầy cần phi tuyến (bài 2 §4).

Ghép lại: cái nào undo được?



$(3, 2) \rightarrow (3, 2, 5) \rightarrow (3, 2)$: khôi phục nguyên vẹn.



$(1, 2, 5)$ và $(1, 2, 99) \rightarrow$ cùng $(1, 2, 3)$. $\det(E \cdot P) = 0$, hạng 2.

	làm gì	undo?
$P \cdot E$	nâng rồi nén	ĐƯỢC: chẳng có gì để mất
$E \cdot P$	nén rồi nâng	KHÔNG: z chết ở bước 1

Thông tin mất ở bước NÉN, không bước nào sau lấy lại được.

Ma trận nâng không phải "nghịch đảo" của ma trận nén.

gzip $\rightarrow$ gunzip hồi được vì nén không mất. JPEG 40% thì resize lên 4K cũng không lấy lại chi tiết. $E \cdot P$ là cái thứ hai.

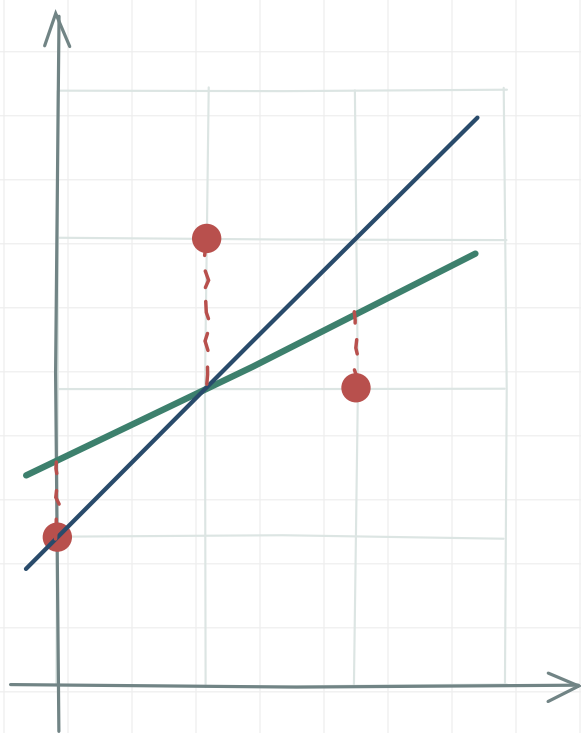
☆ Autoencoder phải HỌC: không thể khôi phục hoàn hảo, chỉ có thể học cách vớt đi thứ ít quan trọng nhất.

"Giải nén" tốt nhất (giả nghịch đảo P^+) cũng chỉ trả $(1, 2) \rightarrow (1, 2, 0)$.

Vô nghiệm? Thay b bằng p

Ba điểm (0,1), (1,3), (2,2) không thẳng hàng: $A \cdot x = b$ vô nghiệm.

$A = [[1,0],[1,1],[1,2]]$ (cột hằng số, cột x), $b = (1, 3, 2)$



$$\hat{x} = (1.5, 0.5)$$

$$p = A \cdot \hat{x} = (1.5, 2, 2.5)$$

← THẲNG HÀNG

$$e = b - p = (-0.5, 1, -0.5)$$

$$\|e\| = 1.2247$$

$$A^T e = (0, 0)$$

— regression $y = 0.5x + 1.5$

— PCA $y = x + 1$

LSQ không "nới lỏng": nó thay b bằng p gần nhất rồi giải ĐÚNG.

Hỏi "gần nhất" $\Rightarrow$ trực giao miễn phí: $e \perp$ mọi cột của A, tức $A^T e = 0$.

Đó chính là phương trình chuẩn $A^T A \hat{x} = A^T b$. Phần e là thứ mô hình KHÔNG học được.

Bẫy: sai số DỌC, không phải vuông góc

	đường	SSE dọc	SSE vuông góc
Regression	$0.5x + 1.5$	1.5 ★	1.2
TLS / PCA	$1.0x + 1.0$	2.0	1.0 ★

"Biết x, đoán y" $\rightarrow$ regression. "Trục chính của đám mây" $\rightarrow$ PCA (bài 9).

$A^T A$ có ĐÚNG null của A $\Rightarrow$ khả nghịch $\Leftrightarrow$ cột độc lập. Nhưng $\kappa(A^T A) = \kappa(A)^2$:

$\kappa = 1e6$ thành $1e12$, mất 12/16 chữ số $\Rightarrow$ dùng lstsq (QR, SVD), đừng lập $A^T A$.

Đây CHÍNH LÀ embedding

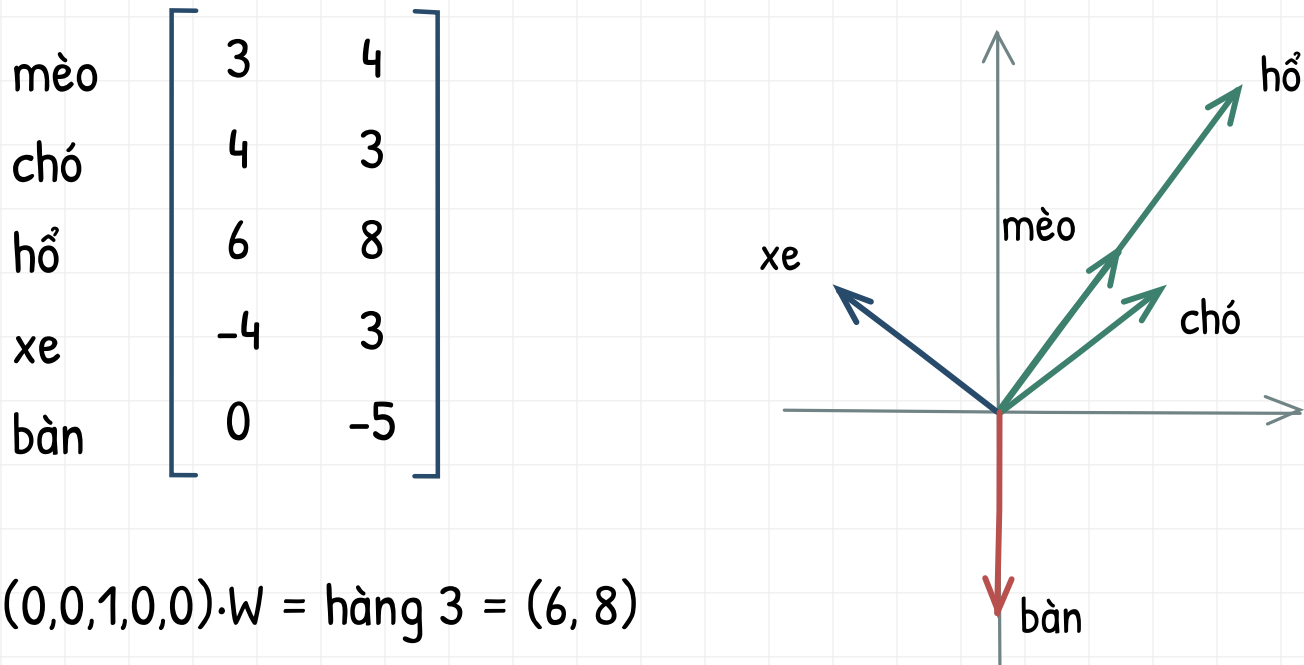
One-hot: "mèo" = (0, ..., 0, 1, 0, ..., 0), 100.000 chiều.

MỌI cặp từ vuông góc: $\cos(\text{mèo}, \text{hổ}) = 0 = \cos(\text{mèo}, \text{cái bàn})$.

Theo bài 1: mọi từ xa lạ với mọi từ như nhau. Vô dụng cho nghĩa.

Embedding = một ma trận nén $\mathbb{R}^{100000} \rightarrow \mathbb{R}^{768}$.

Thu nhỏ: 5 từ, $\mathbb{R}^5 \rightarrow \mathbb{R}^2$. Hàng i = embedding của từ i.



$(0,0,1,0,0) \cdot W = \text{hàng 3} = (6, 8)$

cặp	dot	cosine	đọc là
mèo, hổ	50	1.00	trùng hướng
mèo, chó	24	0.96	rất gần
mèo, xe	0	0.00	không liên quan
mèo, bàn	-20	-0.80	ngược hướng
chó, xe	-7	-0.28	hơi ngược

24 / 0 / -7: đúng ba số bạn tính tay ở bài 1, giờ có tên gọi.

Bây 2: toán viết P.v (cột), thư viện viết x @ W (hàng). Nhìn .shape.

Được quan hệ, mất gì?

$\text{hạng}(W) = 2 \Rightarrow 5 - 2 = 3$ chiều bị nghiền về 0:

$$\text{hổ} - 2 \cdot \text{mèo} = (6, 8) - 2 \cdot (3, 4) = (0, 0)$$

$$5 \cdot \text{chó} + 5 \cdot \text{xe} + 6 \cdot \text{bàn} = (20, 15) + (-20, 15) + (0, -30) = (0, 0)$$

$\cos(\text{mèo}, \text{hổ}) = 1$ CHÍNH XÁC: retrieval dùng cosine không phân biệt nổi.

Mất thông tin không phải bug: là giá của quan hệ ngữ nghĩa.

Huấn luyện embedding = tìm phép nén vút đi thứ ít quan trọng nhất.

Chật đến mức nào? 2000 vector ngẫu nhiên

số chiều d	max cos	cos trung bình
2	1.0000	0.6367
8	0.9910	0.2910
64	0.5657	0.1002
768	0.1761	0.0288

2 chiều: có cặp trùng hướng hoàn toàn. 768 chiều: cặp giống nhau nhất cũng chỉ 0.18, gần như mọi cặp vuông góc. Nhiều hướng gần vuông góc để xếp các nghĩa khác nhau: mặt TỐT của lời nguyên số chiều.

Pigeonhole với $V = 100.000$, $d = 768$

độc lập tuyến tính tối đa trong $\mathbb{R}^{768}$: 768 từ

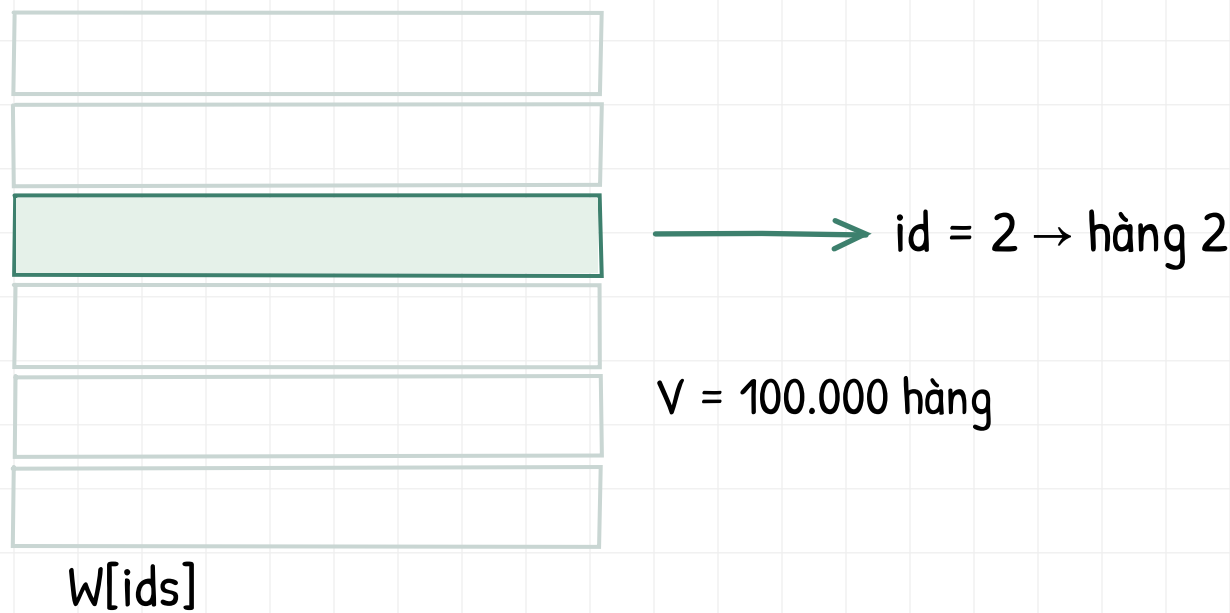
buộc phải phụ thuộc: $100.000 - 768 = 99.232$ từ $\Rightarrow \text{null} \geq 99.232$ chiều

Mô hình vẫn chạy: phần bị vút phần lớn là thứ nó không cần.

nn.Embedding là BẢNG TRA

$(0, 0, 1, 0, 0) \cdot W = 0 \cdot \text{hàng}_0 + 0 \cdot \text{hàng}_1 + 1 \cdot \text{hàng}_2 + 0 \cdot \text{hàng}_3 + 0 \cdot \text{hàng}_4 = \text{hàng}_2$

Nhân one-hot với ma trận = lấy ra MỘT hàng.



V = 100.000, d = 768, batch 32 token	phép nhân-cộng
nhân ma trận đầy đủ	2.457.600.000
số hạng thực sự khác 0	24.576
lãng phí	100.000 lần (99,999 %)

SELECT SUM(vec * CASE WHEN id = :w THEN 1 ELSE 0 END) ← quét toàn bảng
SELECT vec WHERE id = :w ← đi thẳng khoá chính. O(V·d) so với O(d).

Gradient: chỉ hàng được tra mới nhận gradient. 32 token trên 100.000 hàng ⇒ 99,968 % bảng đứng yên mỗi bước ⇒ gradient thưa (sparse).
Về TOÁN nó vẫn là ma trận V → d: hạng, null, mất thông tin y nguyên.

Nối vào AI

Attention

$X (10 \times 768) \cdot W_Q (768 \times 64) \rightarrow (10 \times 64)$; $12 \text{ head} \times 64 = 768$.
Mỗi head cố ý nén 12 lần $\Rightarrow$ chuyên môn hoá. Nén trước, QK^T đo góc sau.

Hạng W_Q

$\text{hạng} \leq 64 \Rightarrow$ mỗi head chỉ thấy không gian con 64 chiều.
null của W_Q có ít nhất $768 - 64 = 704$ chiều.

LoRA

$\Delta W = B \cdot A$: A nén $d \rightarrow r$, B nâng $r \rightarrow d$.
 $\text{hạng}(B \cdot A) \leq r$. Chấp nhận cập nhật hạng thấp, đổi lấy ít tham số.

Autoencoder

Ép qua chỗ hẹp để buộc học cái cốt lõi.
Đúng ví dụ E-P ở tờ 5.

LoRA, $d = 4096$

r	tham số $2 \cdot d \cdot r$	tỉ lệ	ít hơn
1	8.192	0,049 %	2048 lần
4	32.768	0,195 %	512 lần
8	65.536	0,391 %	256 lần
16	131.072	0,781 %	128 lần

Full $d \times d = 16.777.216$ tham số. $r = 8 \Rightarrow$ ít hơn 256 lần.

Tự kiểm, rồi mới sang bài 6

1. 768×100000 đi từ đâu sang đâu? Còn 100000×768 ? Cột là vào hay ra?
 2. Vì sao nén ($m < n$) BẮT BUỘC mất thông tin? Dùng hạng $\leq \min(m, n)$ và hạng + nullity, không dùng ví dụ số.
 3. Vì sao ảnh của E chỉ là mặt phẳng? Phương trình? Một điểm ngoài ảnh?
 4. $P \cdot E = I_2$ nhưng $E \cdot P \neq I_3$: giải thích bằng HÀNH ĐỘNG.
 5. Chứng minh trong 2 dòng: one-hot nhân ma trận ra đúng một hàng.
 6. $V = 100.000$, $d = 768$: tối đa bao nhiêu từ độc lập? Bao nhiêu buộc phụ thuộc? Nghĩa là gì với hai từ $\cos = 1.0$?
 7. One-hot cho $\cos = 0$ mọi cặp; nén xong thì khác 0. Thông tin "mèo gần chó" từ đâu ra, khi phép nén chỉ làm MẤT?
 8. Vì sao chiều cao dễ tách nghĩa hơn? Vì sao không dùng thẳng $d = 100.000$?
- Câu 7 khó nhất: tách "cấu trúc đại số" khỏi "nội dung học được".

Bài tập tay (30 phút, trên giấy)

1. $F = [[1,0,1],[0,1,1]]$: một vector $\neq 0$ bị nghiêng về $(0,0)$? Hạng? null?
2. $G = [[2,0],[0,2],[2,2]]$: ảnh là mặt phẳng nào? So với E: giống hay khác?
3. Tự nghĩ một ma trận 3×5 hạng ĐÚNG 2. Vì sao nó không đạt $\min = 3$?

Mang sang bài 6

Ma trận vuông khả nghịch không mất gì: chỉ viết lại cùng một vector bằng bộ trục khác. "Vector (3, 2)" là câu nói thiếu vế. Thiếu vế nào?

Đáp nhanh

1. $768 \times 100000: \mathbb{R}^{100000} \rightarrow \mathbb{R}^{768}$; 100000×768 : ngược lại. Cột = chiều VÀO.
2. $\text{hạng} \leq m < n \Rightarrow \text{nullity} = n - \text{hạng} \geq n - m > 0$: có hướng bị nghiền về 0.
3. Mọi ảnh là $a \cdot (1,0,1) + b \cdot (0,1,1)$: span của 2 cột $\Rightarrow x + y - z = 0$.
 $(1, 2, 4)$ cho $1 + 2 - 4 = -1 \neq 0 \Rightarrow$ ngoài ảnh.
4. P-E: nâng rồi nén, chẳng mất gì. E-P: nén giết trục z trước, nâng sau không lấy lại được. Không vuông thì không có nghịch đảo hai phía.
5. Cột của W^T = nơi vector cơ sở e_i đáp xuống; one-hot = $e_i \Rightarrow$ ra cột i của W^T = hàng i của W.
6. Tối đa 768 độc lập; 99.232 buộc phụ thuộc. $\cos = 1$: cùng hướng, retrieval bằng cosine không phân biệt nổi.
7. Từ DỮ LIỆU: ma trận nén được học sao cho từ hay đi cùng nhau thì rơi gần nhau. Phép chiếu là cái khuôn; dữ liệu đổ vào khuôn.
8. Nhiều hướng gần vuông góc hơn (tờ 8). $d = 100.000$: đủ chỗ cho mỗi từ một trục riêng, mất áp lực buộc từ gần nghĩa dùng chung hướng (nhận định).

Bài tập tay

1. $F \cdot (1, 1, -1) = (1 - 1, 1 - 1) = (0, 0)$. Hạng 2, null 1 chiều.
2. $G \cdot (a, b) = (2a, 2b, 2a + 2b)$: vẫn mặt phẳng $x + y - z = 0$, GIỐNG E.
 Nhân ma trận với số khác 0 không đổi ảnh, chỉ đổi tọa độ bên trong.
3. Ví dụ $[[1,2,3,4,5],[2,4,6,8,10],[0,1,0,1,0]]$: hàng 2 = $2 \times$ hàng 1
 $\Rightarrow$ chỉ 2 hàng độc lập $\Rightarrow$ hạng 2 < 3.