Linear Algebra

1. Let A be an $m \times n$ matrix with elements a_{ij}

- $(A^T)_{ij} = a_{ji}$

- $((A^T)^T)_{ij} = (A^T)_{ji} = a_{ij}$

Thus, $((A^T)^T)_{ij} = a_{ij}$, so $(A^T)^T = A$.

verify by an example

$$A = \begin{matrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{matrix}$$

The example confirm that $(A^T)^T = A$

$$A^T = \begin{matrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{matrix}$$

$$(A^T)^T = \begin{matrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{matrix} = A$$

2. Let A and B be an $m \times n$ matrices with elements a_{ij} and b_{ij}.

- left side : $(A^T + B^T)_{ij} = (A^T)_{ij} + (B^T)_{ij}$

$= a_{ji} + b_{ji}$

- right side : $(A + B)^T_{ij} = (A + B)_{ji} = a_{ji} + b_{ji}$

Thus, $(A^T + B^T)_{ij} = (A + B)^T_{ij}$ so $A^T + B^T = (A+B)^T$

confirm by an example

let $A = \begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}$, $B = \begin{matrix} 5 & 6 \\ 7 & 8 \end{matrix}$

- left side : $A^T = \begin{matrix} 1 & 3 \\ 2 & 4 \end{matrix}$, $B^T = \begin{matrix} 5 & 7 \\ 6 & 8 \end{matrix}$

so $A^T + B^T = \begin{matrix} 6 & 10 \\ 8 & 12 \end{matrix}$

- right side : $A + B = \begin{matrix} 6 & 10 \\ 8 & 12 \end{matrix}$, so $(A+B)^T = \begin{matrix} 6 & 10 \\ 8 & 12 \end{matrix}$

- Both sides equal, confirming $A^T + B^T = (A+B)^T$

3. Let A be an $n \times n$ square matrix with elements a_{ij}

A matrix is symetric if $M^T = M$

- consider $M = A + A^T$. compute its tranpose

$$M^T = (A + A^T)^T$$

- using resuet from exercise 2

$$(A^T + B^T = (A + B)^T)$$

$$(A + A^T)^T = A^T + (A^T)^T$$

- using result from exercise 1

$((A^T)^T = A)$

- thus $(A + A^T)^T = A^T + A$

$= A + A^T$

$= M$

since $M^T = M$, $A + A^T$ is symetric

example:

let $A = \begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}$

- compute $A^T = \begin{matrix} 1 & 3 \\ 2 & 4 \end{matrix}$

- then $A + A^T = \begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} + \begin{matrix} 1 & 3 \\ 2 & 4 \end{matrix} = \begin{matrix} 2 & 5 \\ 5 & 8 \end{matrix}$

- check transpose $(A + A^T)^T = \begin{matrix} 2 & 5 \\ 5 & 8 \end{matrix}^T = \begin{matrix} 2 & 5 \\ 5 & 8 \end{matrix}$

- since $A + A^T = (A + A^T)^T$, the matrix is symetric

4.

for a tensor X of shape (2, 3, 4), the function len (X) returns the size of the first dimension

- shape of X: (2, 3, 4)

- len (X) = first dimension = 2.

example to confirm

```
import torch

x = torch.zeros((2, 3, 4))
print(len(x))
```

output : 2

5. for a tensor X of arbitrary shape $(d_1, d_2, \ldots, d_n)$, the function len (X) return the size of the first dimension

- shape of X : $(d_1, d_2, \ldots, d_n)$
- len X = first dimension $= d_1$

6. for tensor A of shape (m, n)

with elements a_{ij}, and rows sum

$$s_i = \sum_{j=1}^{n} a_{ij}$$

$$(A / A.sum(axis = 1))_{ij} = \frac{a_{ij}}{s_i}$$

result shape : (m, n)

7. In Manhattan, the distance between two points with coordinates $(x1, y1)$ and $(x2, y2)$ where x represents avenues and y represents streets, is given by the Manhattan distance

$$\text{Distance} = |x2 - x1| + |y2 - y1|$$

example

points (2, 3) and (5, 7)

distance $= |5-2| + |7-3|$

$= 3 + 4$

$= 7$ units (3 avenues + 4 streets)

8. for tensor X of shape (2, 3, 4)

_ sum along axis 0 : shape (3, 4)

_ sum along axis 1 : shape (2, 4)

_ sum along axis 2 : shape (2, 3)

9. for a tensor X $(d_1, d_2, \ldots, d_n)$

the function compute Frobenius norm.

$$\|X\|_F = \sqrt{\sum_{i_1=1}^{d_1} \sum_{i_2=1}^{d_2} \cdots \sum_{i_n=1}^{d_n} |x_{i_1, i_2, \ldots i_n}|^2}$$

output : scalar

10. $A \in R^{2^{10} \times 2^{16}}$, $B \in R^{2^{16} \times 2^5}$, $C \in R^{2^5 \times 2^{14}}$, (16)

compute ABC.

comparison: $(AB)C$ requires less memory

$(2^{15} << 2^{30})$

1. memory foot print

* $(AB)C$:

- AB shape $(2^{10}, 2^5) = (1024, 32)$ elements $= 2^{10} \times 2^5 = 2^{15}$
- $(AB)C$ shape $(2^{10}, 2^{14}) = (1024, 16384)$; elements $= 2^{10} \times 2^{14} = 2^{24}$
- storage : 2^{15} element

* $A(BC)$

- BC shape $(2^{16}, 2^{14}) = (65536, 16384)$ elements $= 2^{16} \times 2^{14} = 2^{30}$
- $A(BC)$ shape $(2^{10}, 2^{14}) = (1024, 16384)$, elements $= 2^{24}$
- storage : 2^{30} elements

2. speed

* (AB) C :

- AB : $2^{10} \times 2^{16} \times 2^{5} = 2^{31}$

- (AB)C : $2^{10} \times 2^{5} \times 2^{14} = 2^{29}$

- total : $2^{31} + 2^{29} \approx 2.5 \times 2^{29}$

* A(BC)

- BC : $2^{16} \times 2^{5} \times 2^{14} = 2^{35}$

- A(BC) : $2^{10} \times 2^{16} \times 2^{14} = 2^{40}$

- total : $2^{35} + 2^{40} \approx 2^{40}$

comparison : (AB)C is faster ($2.5 \times 2^{29} \ll 2^{40}$)

11. for matrices $A \in R^{2^{10} \times 2^{16}}$, $B \in R^{2^{16} \times 2^5}$, $C \in R^{2^5 \times 2^{16}}$

1. speed comparison (AB vs AC^T)

* AB
- shape : $(2^{10}, 2^5) = (1024, 32)$
- flops : $2^{10} \times 2^{16} \times 2^5 = 2^{31}$

* AC^T :
- C^T : shape $(2^{16}, 2^5)$, no computation cost
- AC^T : shape $(2^{10}, 2^5) = (1024, 32)$
- flops : $2^{10} \times 2^{16} \times 2^5 = 2^{31}$

2. if $C = B^T$ without cloning:

- B^T: shape $(2^5, 2^{16})$, same as C; no memory copy (view of B)
- $AC^T = A(B^T)^T = AB$, using $(A^T)^T = A$
- speed: same as AB, 2^{31} flops
- memory: no additional storage for C, as it shares B's memory
- comparison: no speed difference, but $C = B^T$ saves memory by avoid duplication.

12. for matrices $A, B, C \in R^{100 \times 200}$

- construct tensor :

  + stack [A, B, C] along a new axis (default axis 0)

  + tensor shape : (3, 100, 200)

- slice to recover B

  + B is at index 1 of the first axis, 0-based indexing)

  + slide : x [1, :, :] , shape (100, 200)