

Các Phép Tính Numpy, Pytorch và Tensorflow

Transpose và Summation

Dinh-Tiem Nguyen và Quang-Vinh Dinh

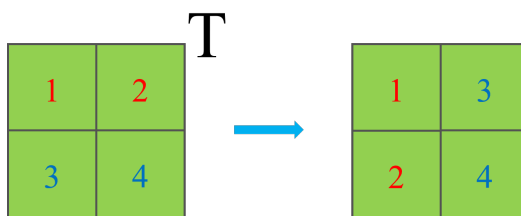
1. Mô tả

a) *Chuyển vị - Transpose*

Chuyển vị (Transpose) là một phép toán quan trọng trong đại số tuyến tính, nó thực hiện thay đổi vị trí của các hàng thành cột và ngược lại trong một ma trận. Ký hiệu của phép toán chuyển vị thường được biểu diễn bằng ký hiệu T hoặc là \top . Nếu A là một ma trận, thì chuyển vị của A được ký hiệu là A^T hoặc A^\top .

Công Thức Chuyển Vị: Nếu A là một ma trận với các phần tử a_{ij} , thì chuyển vị của A , ký hiệu là A^T hay A^\top , có kích thước là số cột của A trở thành số hàng của A và ngược lại. Cụ thể, nếu A có kích thước $m \times n$, thì A^T có kích thước $n \times m$.

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



Hình 1: Minh họa transpose

Ví dụ:

Giả sử có ma trận A như sau:

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

Chuyển vị của A , ký hiệu là A^T , sẽ là:

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

Một số tính chất của phép chuyển vị:

- Chuyển vị của một ma trận chuyển vị lại sẽ cho ra ma trận ban đầu: $(A^T)^T = A$.
- Chuyển vị của tổng hai ma trận bằng tổng chuyển vị của từng ma trận: $(A + B)^T = A^T + B^T$.
- Chuyển vị của tích hai ma trận bằng tích đảo ngược vị trí của chuyển vị từng ma trận: $(AB)^T = B^T A^T$.

Ví dụ:

```

1 # Numpy code
2 import numpy as np
3
4 #Tạo array 2d
5 arr_1 = np.array([[1, 2], [3, 4]])
6 # Chuyển vị array
7 # Cách 1: Sử dụng hàm np.transpose()
8 arr_transposed_1 = np.transpose(arr_1)
9 # Cách 2: Sử dụng toán tử T
10 arr_transposed_2 = arr_1.T
11 # In ra màn hình
12 print("array 1:\n", arr_1)
13 print("array 1 sau khi chuyển vị, cách 1:\n", arr_transposed_1)
14 print("array 1 sau khi chuyển vị, cách 2:\n", arr_transposed_2)

```

```

===== Output =====
array 1:
[[1 2]
 [3 4]]

array 1 sau khi chuyển vị, cách 1:
[[1 3]
 [2 4]]

array 1 sau khi chuyển vị, cách 2:
[[1 3]
 [2 4]]

=====

```

Trong Numpy, chuyển vị của một mảng (array) có thể được thực hiện bằng cách sử dụng hàm **np.transpose()** hoặc toán tử chuyển vị **.T**. Chuyển vị thay đổi vị trí của các hàng thành cột và ngược lại trong array.

Trong Pytorch, chuyển vị của tensor có thể được thực hiện bằng cách sử dụng toán tử chuyển vị **.T**, hàm **torch.t()** hoặc **torch.transpose()**. Chuyển vị thay đổi vị trí của các hàng thành cột và ngược lại trong tensor. Cú pháp:

```

1 #cách 1:
2 torch.t(input)
3
4 #Cách 2
5 torch.transpose(input, dim0, dim1)

```

Kết quả trả về là một tensor mới là kết quả của phép chuyển vị. Trong đó:

- input: Tensor cần được chuyển vị.
- dim0, dim1: Các chiều được chọn để thực hiện chuyển vị.

Ví dụ:

```

1
2 #Pytorch code
3 import torch
4
5 #Tạo tensor 2d
6 tensor_1 = torch.tensor([[1, 2], [3, 4]])
7 # Chuyển vị tensor
8 # Cách 1: Sử dụng hàm torch.t()
9 tensor_transposed_1 = torch.t(tensor_1)
10 # Cách 2: Sử dụng hàm torch.transpose()
11 tensor_transposed_2 = torch.transpose(
    tensor_1, 0, 1)
12 # In ra màn hình
13 print("tensor 1:\n", tensor_1)
14 print("tensor 1 sau khi chuyển vị, cách 1:\n", tensor_transposed_1)
15 print("tensor 1 sau khi chuyển vị, cách 2:\n", tensor_transposed_2)

```

```

===== Output =====
tensor 1:
tensor([[1, 2],
        [3, 4]])

tensor 1 sau khi chuyển vị, cách 1:
tensor([[1, 3],
        [2, 4]])

tensor 1 sau khi chuyển vị, cách 2:
tensor([[1, 3],
        [2, 4]])

=====

```

Trong Tensorflow, không có toán tử chuyển vị **T**, mà chỉ dùng hàm **tf.transpose()**. Cách sử dụng cũng tương tự như Numpy và Pytorch:

Ví dụ:

```
1 # Tensorflow code
2 import tensorflow as tf
3
4 # Tạo tensor 2d
5 tensor_1 = tf.constant([[1, 2], [3, 4]])
6 # Chuyển vị tensor
7 tensor_transposed = tf.transpose(tensor_1)
8
9 # In ra màn hình
10 print("tensor 1:\n", tensor_1)
11 print("tensor 1 sau khi chuyển vị:\n",
      tensor_transposed)
```

```
===== Output =====
tensor 1:
tf.Tensor(
[[1 2]
 [3 4]], shape=(2, 2), dtype=int32)

tensor 1 sau khi chuyển vị:
tf.Tensor(
[[1 3]
 [2 4]], shape=(2, 2), dtype=int32)
=====
```

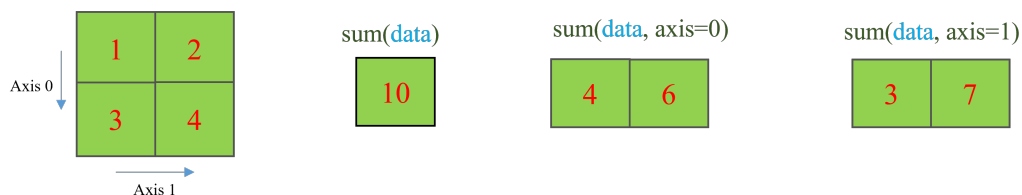
b) *Summation*

Trong thư viện Numpy, hàm **np.sum()** được sử dụng để tính tổng của các phần tử trong mảng (array). Hàm này có thể được áp dụng trên các mảng 1D, 2D hoặc có số chiều cao hơn. Cú pháp:

```
1 np.sum(a, axis=None, dtype=None, keepdims=False, initial=0, where=True)
```

Trong đó:

- a: Mảng đầu vào.
- axis: (Tùy chọn) Chiều hoặc các chiều trên đó tổng sẽ được thực hiện. Mặc định là None, tức là tổng của tất cả các phần tử trong mảng.
- dtype: (Tùy chọn) Kiểu dữ liệu của kết quả.
- keepdims: (Tùy chọn) Nếu là True, giữ chiều của mảng kết quả (nếu có) giống với chiều của mảng đầu vào.
- initial: (Tùy chọn) Giá trị khởi tạo cho tổng.
- where: (Tùy chọn) Một mảng Boolean chỉ định vị trí các phần tử được sử dụng trong phép toán.



Hình 2: Minh họa summation

Trong ví dụ sau, **np.sum()** được sử dụng để tính tổng của mảng array. Tham số sử dụng ở đây là mặc định khi tính tổng các phần tử trong toàn bộ array, sử dụng tham số axis để tính tổng theo cột hoặc hàng. Kết quả được in ra màn hình bao gồm tổng của tất cả các phần tử, tổng theo cột, và tổng theo hàng của mảng.

```

1 # Numpy code
2 import numpy as np
3
4 # Tạo mảng 2D
5 arr = np.array([[1, 2, 3],
6                 [4, 5, 6]])
7
8 # Tính tổng của tất cả các phần tử trong m
   ảng
9 total_sum = np.sum(arr)
10
11 # Tính tổng theo cột (theo chiều dọc)
12 column_sum = np.sum(arr, axis=0)
13
14 # Tính tổng theo hàng (theo chiều ngang)
15 row_sum = np.sum(arr, axis=1)
16
17 # In ra màn hình
18 print("Mảng:\n", arr)
19 print("Tổng của tất cả các phần tử trong m
   ảng:\n", total_sum)
20 print("Tổng theo cột:\n", column_sum)
21 print("Tổng theo hàng:\n", row_sum)

```

```

===== Output =====
Mảng:
[[1 2 3]
 [4 5 6]]
Tổng của tất cả các phần tử trong mảng:
21
Tổng theo cột:
[5 7 9]
Tổng theo hàng:
[ 6 15]
=====

```

Trong Pytorch, chúng ta tính sum tương tự như trong Numpy, ví dụ:

```

1 #Pytorch code
2 import torch
3
4 # Tạo tensor 2d
5 tensor_1 = torch.tensor([[1, 2], [3, 4]])
6 # Tính tổng tensor
7 tensor_sum = torch.sum(tensor_1)
8 # Tính tổng tensor theo cột axis = 0
9 tensor_sum_axis_0 = torch.sum(tensor_1,
   axis=0)
10 # Tính tổng tensor theo hàng axis = 1
11 tensor_sum_axis_1 = torch.sum(tensor_1,
   axis=1)
12
13 # In ra màn hình
14 print("tensor:\n", tensor_1)
15 print("Tổng các phần tử trong tensor:\n",
   tensor_sum)
16 print("Tổng theo cột:\n",
   tensor_sum_axis_0)
17 print("Tổng theo hàng:\n",
   tensor_sum_axis_1)

```

```

===== Output =====
tensor:
tensor([[1, 2],
        [3, 4]])
Tổng các phần tử trong tensor:
tensor(10)
Tổng theo cột:
tensor([4, 6])
Tổng theo hàng:
tensor([3, 7])
=====

```

Ví dụ tính sum trong Tensorflow:

```

1 # Tensorflow code
2 import tensorflow as tf
3
4 # Tạo tensor 2d
5 tensor_1 = tf.constant([[1, 2], [3, 4]])
6 # Tính tổng tensor
7 tensor_sum = tf.reduce_sum(tensor_1)
8 # Tính tổng tensor theo cột axis = 0
9 tensor_sum_axis_0 = tf.reduce_sum(tensor_1
    , axis=0)
10 # Tính tổng tensor theo hàng axis = 1
11 tensor_sum_axis_1 = tf.reduce_sum(tensor_1
    , axis=1)
12
13 # In ra màn hình
14 print("tensor:\n", tensor_1)
15 print("Tổng các phần tử trong tensor:\n",
    tensor_sum)
16 print("Tổng theo cột:\n",
    tensor_sum_axis_0)
17 print("Tổng theo hàng:\n",
    tensor_sum_axis_1)

```

```

===== Output =====
tensor:
  tf.Tensor(
[[1 2]
 [3 4]], shape=(2, 2), dtype=int32)

Tổng các phần tử trong tensor:
  tf.Tensor(10, shape=(), dtype=int32)

Tổng theo cột:
  tf.Tensor([4 6], shape=(2,), dtype=int32)

Tổng theo hàng:
  tf.Tensor([3 7], shape=(2,), dtype=int32)

=====

```

2. Bài tập

Câu 1: Viết chương trình tạo hai Numpy array, Pytorch tensor, Tensorflow tensor với các giá trị số nguyên ngẫu nhiên trong khoảng $[-10, 10]$ với kích thước (3, 4). Sau đó chuyển vị array, tensor thứ 2 và thực hiện phép nhân matrix multiplication. Lưu ý: sử dụng seed=2024

Câu 2: Viết chương trình tạo một Numpy array, Pytorch tensor, Tensorflow tensor với các giá trị số nguyên ngẫu nhiên trong khoảng $[-10, 10]$ với kích thước (3, 3). Sau đó hãy tính tổng của toàn bộ tensor, array, tiếp theo tính tổng theo chiều dọc, chiều ngang. Lưu ý: sử dụng seed=2024

3. Đáp án

Đáp án sẽ được gửi cho các bạn vào khoảng 8h tối trên group Code.