**Phần 2\_7\_Lớp\_Phép biến đổi hình dạng dữ liệu**

**Notes:**

* *Về format cấu trúc soạn như sau:*
  + *Lý thuyết…*
  + *Bộ code mấu/ ví dụ …*
  + *Ứng dụng (nếu có)...*
* *Mems làm nhớ note tên để mn dễ contact*

### **I. Nội dung chính**

Về các phép tính toán (lớp tính toán) để xây dựng mô hình học sâu.

* + (7) Lớp/Phép biến đổi hình dạng dữ liệu, hoán vị các chiều trong bản đồ đặc trưng và chia tách bản đồ đặc trưng: **Reshape/View/Flatten**; **Transpose/Permute; Split/Chunk**
    - Nguyên lý làm việc
    - Ràng buộc về hình dạng và kích thước của các bản đồ đặc trưng ở đầu vào và đầu ra

### **II. Nội dung biên soạn chi tiết**

Nguyên lý làm việc: Các lớp này thực hiện các phép biến đổi trên hình dạng của dữ liệu mà không thay đổi nội dung của chúng.

* Reshape/View/Flatten thay đổi hình dạng của tensor mà không làm thay đổi tổng số phần tử.
* Transpose/Permute hoán đổi các chiều của tensor.
* Split/Chunk chia tensor thành nhiều tensor nhỏ hơn.

Ràng buộc về hình dạng và kích thước của các bản đồ đặc trưng ở đầu vào và đầu ra: Kích thước của tensor đầu ra phụ thuộc vào cách thức thực hiện phép biến đổi. Ví dụ, phép Reshape sẽ tạo ra tensor có kích thước mới mà tổng số phần tử không đổi.

**Code VD:**

import torch

# Tạo một tensor ngẫu nhiên

x = torch.randn(2, 3, 4)

print(f"Original shape: {x.shape}")

# Sử dụng Reshape/View để thay đổi hình dạng tensor

x\_reshaped = x.view(6, 4)

print(f"Reshaped tensor shape: {x\_reshaped.shape}")

# Sử dụng Transpose để hoán đổi các chiều

x\_transposed = x.transpose(0, 1)

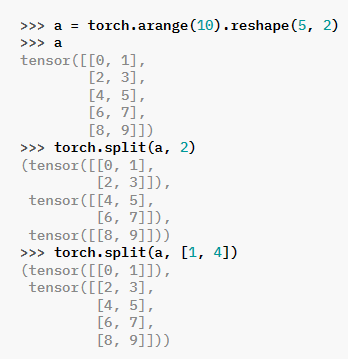
print(f"Transposed tensor shape: {x\_transposed.shape}")

# Sử dụng Split để chia tensor

chunks = x.split(1, dim=0)

for i, chunk in enumerate(chunks):

print(f"Chunk {i+1} shape: {chunk.shape}")



Flatten:

