**Phần 2\_6\_Lớp tổng hợp dữ liệu Concat, ElementWise-Add**

**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*

Mem: Nghia Pham

### **I. Nội dung chính**

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

* + (6) Lớp tổng hợp dữ liệu: **Concat**, **ElementWise-Add**
    - 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**

#### **1. Concat (Nối):**

* Nguyên lý làm việc: Phép nối Concat trong học sâu được sử dụng để nối các tensor (hoặc bản đồ đặc trưng) theo một chiều cụ thể. Điều này giúp kết hợp thông tin từ các tensor khác nhau.
* Ràng buộc về hình dạng và kích thước: Để thực hiện phép nối, tất cả các tensor cần phải có cùng kích thước trên tất cả các chiều, ngoại trừ chiều mà chúng được nối. Ví dụ, nếu bạn có hai tensor 3D có kích thước (32, 10, 10) và (64, 10, 10), bạn có thể nối chúng theo chiều đầu tiên để tạo ra một tensor mới có kích thước (96, 10, 10).

**Code VD:**

import torch

# Tạo hai tensor 3D

tensor1 = torch.rand((32, 10, 10))

tensor2 = torch.rand((64, 10, 10))

# Nối chúng theo chiều đầu tiên

result = torch.cat((tensor1, tensor2), dim=0)

print(result.shape) # Kết quả sẽ là: torch.Size([96, 10, 10])

#### **2. ElementWise-Add (Cộng từng phần tử):**

* Nguyên lý làm việc: Phép ElementWise-Add trong học sâu thực hiện phép cộng giữa các tensor (hoặc bản đồ đặc trưng) theo từng phần tử tương ứng. Điều này giúp kết hợp thông tin từ các tensor khác nhau mà không làm thay đổi kích thước của chúng.
* Ràng buộc về hình dạng và kích thước: Để thực hiện phép ElementWise-Add, tất cả các tensor cần phải có cùng kích thước trên tất cả các chiều. Ví dụ, nếu bạn có hai tensor 3D có kích thước (32, 10, 10), bạn có thể thực hiện phép ElementWise-Add để tạo ra một tensor mới cũng có kích thước (32, 10, 10).

Code VD: