**Phần 2\_1\_Lớp tuyến tính**

By **Nguyễn Phạm Ngọc Quý**

**Mục lục**

[**I. Nội dung chính**](#_pw0fe9hp5sgb) **1**

[**II. Nội dung biên soạn chi tiết**](#_e2yd5dnxchkc) **1**

[1. Lớp kết nối đầy đủ (Fully-Connected Layer)](#_25aa8nejbhy) 1

[2. Lớp tích chập (Convolution Layer)](#_1esdsxgr9j9k) 2

[3. Các biến thể của Convolution](#_t9xdx6enajqe) 3

### **I. Nội dung chính**

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

* + (1) Lớp tuyến tính: Kết nối đầy đủ (**FC**), tích chập (**Convolution**) và các biến thể của Convolution như **Separable** (space vs channel), **DepthWise**, **PointWise, Transposed-Convolution.**

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

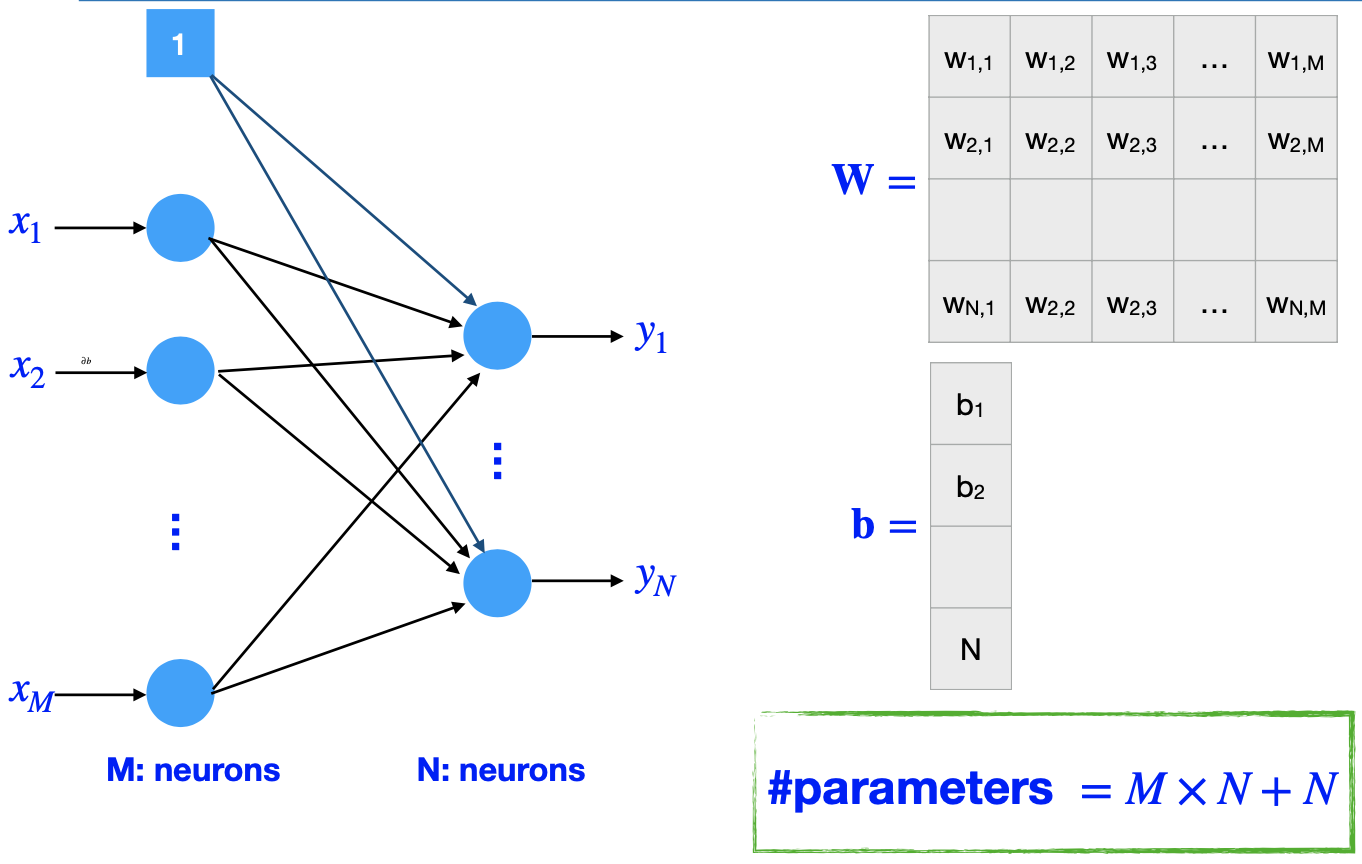
#### **1. Lớp kết nối đầy đủ (Fully-Connected Layer)**

* **Nguyên lý làm việc:** Tầng kết nối đầy đủ (FC) nhận đầu vào là các dữ liệu đã được làm phẳng, mà mỗi đầu vào đó được kết nối đến tất cả neuron. Trong mô hình mạng CNNs, các tầng kết nối đầy đủ thường được tìm thấy ở cuối mạng và được dùng để tối ưu hóa mục tiêu của mạng ví dụ như độ chính xác của lớp.
* **Công thức liên hệ giữa đầu vào và đầu ra:**
* y là ma trận đầu ra
* x là ma trận đầu vào
* W là ma trận trọng số. Các trọng số của ma trận trọng số được học theo phương pháp ***gradient descent***.
* b là hằng số, được sử dụng để điều chỉnh độ dốc của hàm mất mát.

Để áp dụng gradient descent ta cần tính được đạo hàm của các hệ số W và bias b với hàm loss function. Đối với back-propagation, update weight:

* là learning rate
* **Số tham số có thể học:** phụ thuộc vào số lượng neuron trong mỗi lớp.
  + is connected to every ⇒ dense connectivity
  + Each uses different set of and

Ví dụ: W có kích thước NxM, b có kích thước Nx1 thì số tham số **= M x N + N**



* **Độ rộng của vùng nhận thức (receptive fields):** Không có khái niệm vùng nhận thức cụ thể, mỗi neuron ở lớp này nhận thông tin từ toàn bộ input.
* **Hình dạng kích thước của bản đồ đặc trưng đầu vào và đầu ra:** Không có bản đồ đặc trưng, chỉ có vector đầu vào và đầu ra.

#### **2. Lớp tích chập (Convolution Layer)**

* **Nguyên lý làm việc:** Tầng tích chập (CONV) sử dụng các bộ lọc (kernel) để thực hiện phép tích chập khi đưa chúng đi qua đầu vào theo các chiều của nó. Các siêu tham số của các bộ lọc này bao gồm kích thước bộ lọc F và độ trượt (stride) S.
  + ***Padding:*** 
    - Half padding size =
    - Full padding size =
  + ***Stride:*** Stride > 1 có thể được sử dụng để giảm output feature map => giảm tính toán ở các lớp kế tiếp. Khi đó, ta chỉ thực hiện phép tính convolution trên các phần tử

Công thức tổng quát cho phép tính convolution của ma trận X kích thước m x n với kernel kích thước k x k, stride = s, padding = p ra ma trận Y kích thước:

Các bước thực hiện tính toán:

1. Xoay kernel 180°, làm phẳng nó thành một vector
2. Padding zero cho input, allocate output buffer.
3. Align the kernel window vào phía trái trên cùng của input.
4. Trích xuất sub-image trùng với kernel
5. Thực hiện phép nhân vô hướng với kernel đã làm phẳng
6. Điền kết quả vào output
7. Nếu tất cả các giá trị của output đã được điền thì kết thúc, không thì dịch kernel window sang vị trí kế tiếp của input rồi thực hiện bước 4.

* **Công thức liên hệ giữa đầu vào và đầu ra:**

(Chú ý: Dấu “\*” không phải là nhân ma trận)

* X là ma trận đầu vào (image hoặc feature map)
* Y là ma trận đầu ra (feature map hoặc activation map)
* W là ma trận trọng số (Filter's kernel)

\*\*Convolution:

\*\*Cross-correlation:

* **Số tham số có thể học:** được xác định bởi kích thước của kernel và số lượng filters.
  + is connected to pixels in a local region of X ⇒ sparse connectivity
  + all share the same set of parameters: and b ⇒ parameter sharing

=> Số lượng tham số = **k x k x Số lượng filters + Số lượng filters**

(k là kích thước của filter).

Nếu input là một tensor, kernel có kích thước k x k x d thì số lượng tham số = Số lượng kernel \* (k\*k\*d + 1).

* **Độ rộng của vùng nhận thức (receptive fields):** Vùng nhận thức của mỗi neuron được xác định bởi kích thước của kernel. Ví dụ, một neuron trong lớp tích chập sử dụng kernel 3x3 sẽ có vùng nhận thức 3x3.
* **Hình dạng kích thước của bản đồ đặc trưng đầu vào và đầu ra:** Kích thước của bản đồ đặc trưng đầu ra phụ thuộc vào kích thước kernel, padding và stride. Ví dụ, sử dụng kernel 3x3 với stride là 1 và không sử dụng padding sẽ giảm kích thước đầu ra đi 2 hàng và cột so với đầu vào.

#### **3. Các biến thể của Convolution**

* **Separable convolution:** Phân tách convolution thành hai giai đoạn riêng biệt: convolution không gian (spatial convolution) và convolution theo kênh (channel-wise convolution), giúp giảm số lượng tham số.
  + Số tham số được giảm đi so với lớp tích chập thông thường vì việc chia sẻ tham số giữa hai giai đoạn.
  + Công thức liên hệ giữa đầu vào và đầu ra: Sử dụng phép tích chập theo hai giai đoạn: convolution không gian và convolution theo kênh.
* **Depthwise convolution:** Thực hiện convolution trên từng kênh đầu vào một cách độc lập, giữa không gian không thay đổi.
  + Số tham số có thể học: Số lượng tham số giảm đi đáng kể so với lớp tích chập thông thường vì chỉ có một kernel cho mỗi kênh đầu vào.
  + Công thức liên hệ giữa đầu vào và đầu ra: Sử dụng phép tích chập trên từng kênh đầu vào một cách độc lập.
* **Pointwise convolution:** Convolution với kernel kích thước 1x1, thường được sử dụng để kết hợp thông tin giữa các kênh.
  + Số tham số có thể học: Số lượng tham số giảm đi đáng kể vì chỉ có kernel kích thước nhỏ.
  + Độ rộng của vùng nhận thức: Không có khái niệm vùng nhận thức cụ thể, do là convolution 1x1.
  + Hình dạng kích thước của bản đồ đặc trưng đầu vào và đầu ra: Kích thước đầu ra phụ thuộc hoàn toàn vào kích thước đầu vào và số lượng filters.
  + Công thức liên hệ giữa đầu vào và đầu ra: Sử dụng phép tích chập 1x1.
* **Transposed convolution:** Tăng kích thước bản đồ đặc trưng đầu ra bằng cách sử dụng phép tích chập ngược lại.
  + Số tham số có thể học: Tăng lên so với lớp tích chập thông thường do việc tăng kích thước đầu ra.
  + Độ rộng của vùng nhận thức: Tùy thuộc vào kích thước kernel và stride của phép tích chập ngược lại.

Được sử dụng trong các mô hình GANs hoặc trong các tác vụ up-sampling, tức là tăng kích thước của bản đồ đặc trưng.