**Phần 2\_2\_Lớp pooling**

**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*

**Mục lục**

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

[**II. Nội dung biên soạn chi tiết**](#_ahfts7xa5m86) **1**

[1. Tổng quan về lớp gộp (pooling)](#_9940akox4u4j) 1

[a. Ưu điểm của lớp gộp:](#_vdhrs8nmrndb) 2

[b. Nhược điểm của lớp gộp:](#_2sf2ik58nw4b) 2

[2. Max pooling](#_fj584z8ixsc0) 3

[3. Average pooling](#_vmsl1ym5ubc) 4

[4. global pooling](#_6ejnnjetgjpf) 6

### **I. Nội dung chính**

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

* + Lớp pooling: **max-pooling**, **average-pooling**, **global-average-pooling**
    - Nguyên lý làm việc
    - Độ rộng của vùng nhận thức (receptive fields)
    - Hình dạng kích thước của bản đồ đặc trưng đầu vào và đầu ra

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

#### **1. Tổng quan về lớp gộp (pooling)**

Trong các mạng thần kinh tích chập (CNN), lớp gộp (pooling) là một loại lớp phổ biến thường được thêm vào sau các lớp tích chập. Lớp gộp được sử dụng để giảm kích thước không gian (tức là chiều rộng và chiều cao) của bản đồ đối tượng, trong khi vẫn duy trì độ sâu (tức là số lượng kênh).

Lớp gộp hoạt động bằng cách chia bản đồ tính năng đầu vào thành một tập hợp các vùng không chồng chéo, được gọi là vùng gộp. Sau đó, mỗi vùng gộp được chuyển đổi thành một giá trị đầu ra duy nhất, thể hiện sự hiện diện của một tính năng cụ thể trong vùng đó. Các loại hoạt động gộp phổ biến nhất là gộp tối đa và gộp trung bình.

Trong max pooling , giá trị đầu ra cho từng vùng gộp chỉ đơn giản là giá trị tối đa của các giá trị đầu vào trong vùng đó. Điều này có tác dụng bảo tồn các đặc điểm nổi bật nhất trong từng khu vực tổng hợp, đồng thời loại bỏ những thông tin ít liên quan hơn. Tổng hợp tối đa thường được sử dụng trong CNN cho các nhiệm vụ nhận dạng đối tượng, vì nó giúp xác định các đặc điểm đặc biệt nhất của đối tượng, chẳng hạn như các cạnh và góc của nó.

Trong tính năng average pooling, giá trị đầu ra cho từng vùng gộp là giá trị trung bình của các giá trị đầu vào trong vùng đó. Điều này có tác dụng lưu giữ nhiều thông tin hơn so với gộp chung tối đa, nhưng cũng có thể làm loãng các tính năng nổi bật nhất. Nhóm trung bình thường được sử dụng trong CNN cho các tác vụ như phân đoạn hình ảnh và phát hiện đối tượng, trong đó yêu cầu thể hiện đầu vào chi tiết hơn.

Các lớp gộp thường được sử dụng kết hợp với các lớp tích chập trong CNN, với mỗi lớp gộp làm giảm kích thước không gian của bản đồ đối tượng, trong khi các lớp tích chập trích xuất các đối tượng ngày càng phức tạp từ đầu vào. Các bản đồ tính năng thu được sau đó được chuyển đến một lớp được kết nối đầy đủ, lớp này thực hiện nhiệm vụ phân loại hoặc hồi quy cuối cùng.

##### **a. Ưu điểm của lớp gộp:**

Giảm kích thước: Ưu điểm chính của các lớp gộp là chúng giúp giảm kích thước không gian của bản đồ đối tượng. Điều này làm giảm chi phí tính toán và cũng giúp tránh việc trang bị quá mức bằng cách giảm số lượng tham số trong mô hình.

Tính bất biến dịch (Translation invariance): Các lớp gộp cũng hữu ích trong việc đạt được tính bất biến dịch trong bản đồ đặc trưng. Điều này có nghĩa là vị trí của một đối tượng trong ảnh không ảnh hưởng đến kết quả phân loại vì các đặc điểm giống nhau được phát hiện bất kể vị trí của đối tượng.

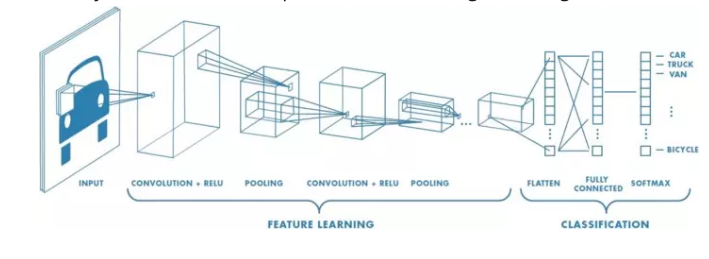
Lựa chọn tính năng: Các lớp gộp cũng có thể giúp chọn các tính năng quan trọng nhất từ đầu vào, vì việc gộp tối đa sẽ chọn các tính năng nổi bật nhất và việc gộp chung trung bình sẽ lưu giữ nhiều thông tin hơn.

##### **b. Nhược điểm của lớp gộp:**

Mất thông tin: Một trong những nhược điểm chính của các lớp gộp là chúng loại bỏ một số thông tin khỏi bản đồ tính năng đầu vào, điều này có thể quan trọng đối với nhiệm vụ phân loại hoặc hồi quy cuối cùng.

Làm mịn quá mức: Các lớp gộp cũng có thể gây ra tình trạng làm mịn quá mức các bản đồ đặc trưng, điều này có thể dẫn đến mất một số chi tiết chi tiết quan trọng đối với nhiệm vụ phân loại hoặc hồi quy cuối cùng.

Điều chỉnh siêu tham số: Các lớp gộp cũng đưa ra các siêu tham số như kích thước của vùng gộp và bước tiến, cần được điều chỉnh để đạt được hiệu suất tối ưu. Việc này có thể tốn thời gian và đòi hỏi kiến thức chuyên môn về xây dựng mô hình.



Hoạt động pooling bao gồm việc trượt bộ lọc hai chiều qua từng kênh của bản đồ đối tượng và tóm tắt các đối tượng nằm trong vùng được bộ lọc bao phủ.

Đối với bản đồ đặc trưng có kích thước Nh x Nw x Nc, kích thước đầu ra thu được sau pooling layer là

(Nh - f + 1)/s \* (Nw - f + 1)/s \* Nc

-> Nh - chiều cao của feature map

-> Nw - chiều rộng của feature map

-> Nc - số lượng kênh trong feature map

-> f - kích thước của bộ lọc

-> s - stride length

Kiến trúc mô hình CNN phổ biến là có một số lớp tích chập (convolution) và lớp gộp được xếp chồng lên nhau.

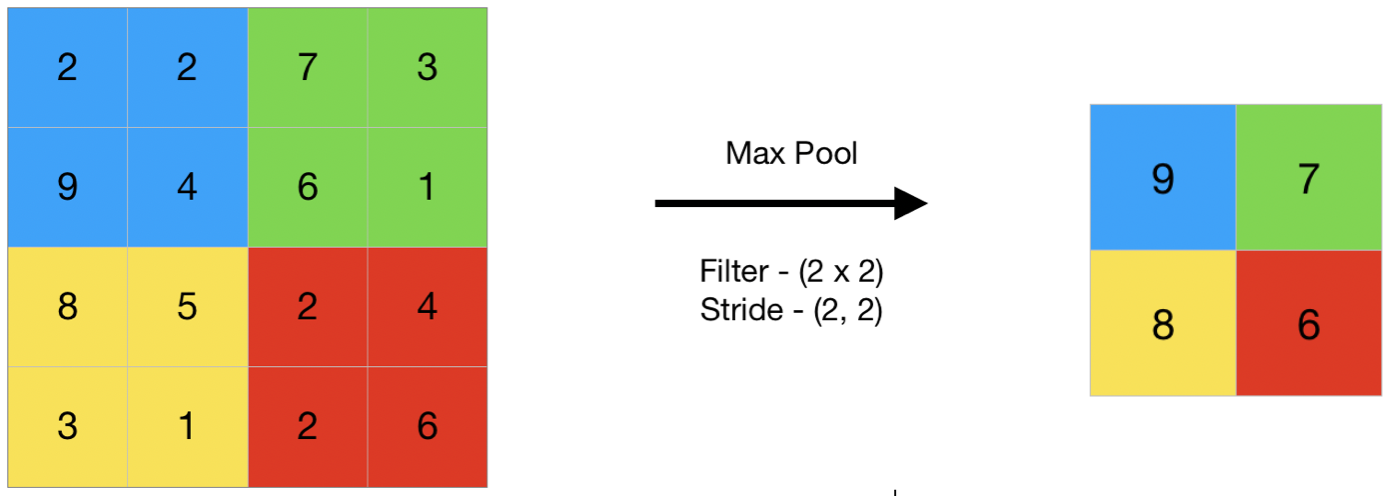
Tại sao nên sử dụng pooling layer?

Các pooling layer được sử dụng để giảm kích thước của bản đồ đối tượng. Do đó, nó làm giảm số lượng tham số cần tìm hiểu và số lượng tính toán được thực hiện trong mạng.

Lớp pooling layer tóm tắt các đối tượng có trong một vùng của bản đồ đối tượng được tạo bởi lớp tích chập. Vì vậy, các hoạt động tiếp theo được thực hiện trên các đối tượng được tóm tắt thay vì các đối tượng được định vị chính xác do lớp tích chập tạo ra. Điều này làm cho mô hình trở nên chắc chắn hơn trước những thay đổi về vị trí của các đặc điểm trong ảnh đầu vào.

#### **2. Max pooling**

**Max pooling** là hoạt động gộp để chọn phần tử tối đa từ vùng của bản đồ đối tượng được bao phủ bởi bộ lọc. Do đó, đầu ra sau lớp tổng hợp tối đa sẽ là bản đồ đặc trưng chứa các đặc điểm nổi bật nhất của bản đồ đặc trưng trước đó.



**code:**

**import** numpy as np

**from** keras.models **import** Sequential

**from** keras.layers **import** MaxPooling2D

# define input image

image **=** np.array([[2, 2, 7, 3],

[9, 4, 6, 1],

[8, 5, 2, 4],

[3, 1, 2, 6]])

image **=** image.reshape(1, 4, 4, 1)

# define model containing just a single max pooling layer

model **=** Sequential(

[MaxPooling2D(pool\_size **=** 2, strides **=** 2)])

# generate pooled output

output **=** model.predict(image)

# print output image

output **=** np.squeeze(output)

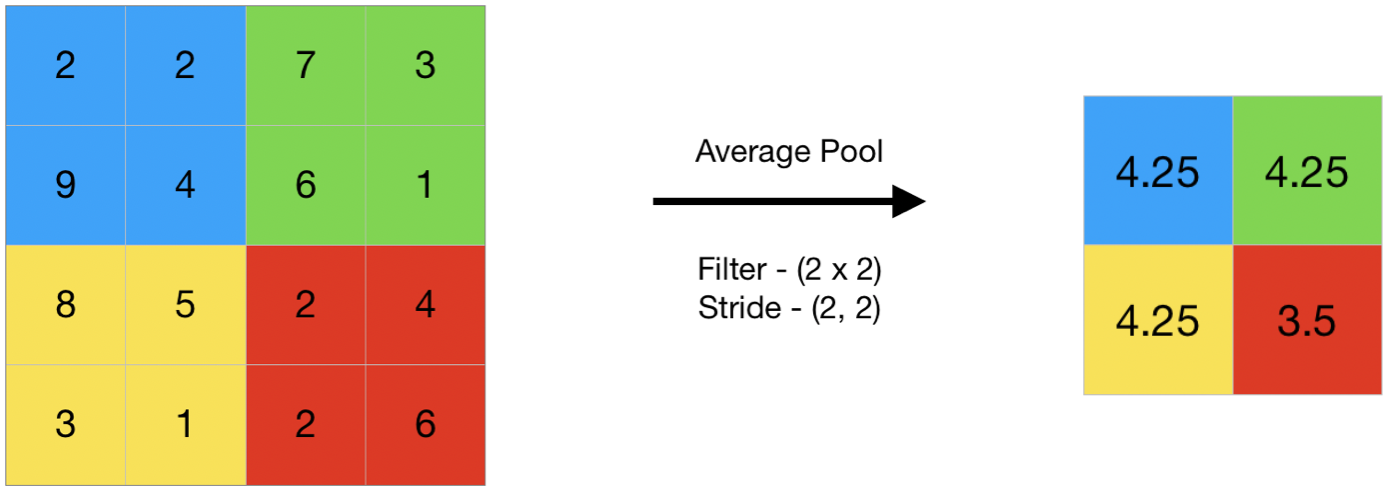
print(output)

**Output:** [[9. 7.]

[8. 6.]]

#### **3. Average pooling**

Average pooling tính toán mức trung bình của các phần tử có trong vùng feature map được bao phủ bởi bộ lọc. Do đó, trong khi việc gộp tối đa mang lại tính năng nổi bật nhất trong một patch cụ thể của bản đồ tính năng, thì việc gộp trung bình cho tính năng trung bình có trong một patch.



**code:**

**import** numpy as np

**from** keras.models **import** Sequential

**from** keras.layers **import** AveragePooling2D

# define input image

image **=** np.array([[2, 2, 7, 3],

[9, 4, 6, 1],

[8, 5, 2, 4],

[3, 1, 2, 6]])

image **=** image.reshape(1, 4, 4, 1)

# define model containing just a single average pooling layer

model **=** Sequential(

[AveragePooling2D(pool\_size **=** 2, strides **=** 2)])

# generate pooled output

output **=** model.predict(image)

# print output image

output **=** np.squeeze(output)

print(output)

**Output:**  [[4.25 4.25]

[4.25 3.5 ]]

#### **4. global pooling**

Việc gộp chung sẽ giảm mỗi kênh trong feature map xuống một giá trị duy nhất. Do đó, feature map Nh \* Nw \* Nc được giảm xuống còn bản đồ đặc trưng 1 x 1 x nc. Điều này tương đương với việc sử dụng bộ lọc kích thước Nh xNnw tức là kích thước của bản đồ đối tượng.

Hơn nữa, nó có thể là tổng hợp global max pooling hoặc tổng hợp global average pooling

**code:**

**import** numpy as np

**from** keras.models **import** Sequential

**from** keras.layers **import** GlobalMaxPooling2D

**from** keras.layers **import** GlobalAveragePooling2D

# define input image

image **=** np.array([[2, 2, 7, 3],

[9, 4, 6, 1],

[8, 5, 2, 4],

[3, 1, 2, 6]])

image **=** image.reshape(1, 4, 4, 1)

# define gm\_model containing just a single global-max pooling layer

gm\_model **=** Sequential(

[GlobalMaxPooling2D()])

# define ga\_model containing just a single global-average pooling layer

ga\_model **=** Sequential(

[GlobalAveragePooling2D()])

# generate pooled output

gm\_output **=** gm\_model.predict(image)

ga\_output **=** ga\_model.predict(image)

# print output image

gm\_output **=** np.squeeze(gm\_output)

ga\_output **=** np.squeeze(ga\_output)

print("gm\_output: ", gm\_output)

print("ga\_output: ", ga\_output)

**Output:**

gm\_output: 9.0

ga\_output: 4.0625

**giải thích**

max ([2, 2, 7, 3],

[9, 4, 6, 1], = 9

[8, 5, 2, 4],

[3, 1, 2, 6]])

average ([2, 2, 7, 3],

[9, 4, 6, 1], = (2+2+7+3…..+3+1+2+6)/16 = 4.0625

[8, 5, 2, 4],

[3, 1, 2, 6]])

link tham khảo (<https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/>)