

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP.HCM

KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN MÔN HỌC SÂU

**Nhận diện và phân lớp phương tiện giao thông
(Vehicle Detection and Classification)**

Ngành: Công Nghệ Thông Tin

Chuyên ngành: Trí Tuệ Nhân Tạo

Giảng viên hướng dẫn: ThS. Lê Nhật Tùng

Sinh viên thực hiện: Trịnh Quốc Trọng 2180608444

Nguyễn Minh Thắng 2180608048

Hồ Phát Đạt 2180607416

TP. Hồ Chí Minh, 04/2025

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting or typing. There are no margins, text, or other markings on the page.

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn sâu sắc đến Thầy Vũ Thanh Hiền, người đã hướng dẫn và truyền đạt cho chúng em những kiến thức quý báu trong suốt quá trình thực hiện bài báo cáo này. Thầy không chỉ cung cấp cho chúng em những kiến thức chuyên môn mà còn dành thời gian giải đáp những khó khăn, thắc mắc mà chúng em gặp phải. Sự chỉ dẫn tận tình và sự tin tưởng của Thầy chính là nguồn động lực lớn giúp chúng em hoàn thành bài báo cáo này.

Chúng em hiểu rằng, với thời gian thực hiện có hạn, bài báo cáo chắc chắn còn tồn tại nhiều thiếu sót và chưa thể đạt đến mức hoàn thiện. Chúng em rất mong nhận được những góp ý từ Thầy để nhóm có thể cải thiện và học hỏi thêm trong tương lai. Sự đóng góp của Thầy không chỉ giúp bài báo cáo được tốt hơn mà còn là bài học giúp chúng em tích lũy kinh nghiệm cho những công việc sau này.

Một lần nữa, chúng em xin gửi lời tri ân chân thành đến Thầy và hy vọng sẽ tiếp tục nhận được sự giúp đỡ, chỉ bảo từ Thầy trong những chặng đường học tập tiếp theo.

Trân trọng cảm ơn!

Trịnh Quốc Trọng

Chu Tiến Bình

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI.....	1
1.1. Tên đề tài	1
1.2. Giới thiệu đề tài	1
1.3. Phạm vi nghiên cứu.	1
1.4. Ý nghĩa thực tiễn của đề tài:.....	2
1.7. Các mô hình, công cụ được sử dụng trong đề tài:	3
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	4
2.1. Mạng neural hồi quy.	4
2.1.1. Kiến trúc của một mạng RNN truyền thống.....	4
2.1.2. Ứng dụng của RNNs.	5
2.1.3. Hàm mất mát.	7
2.1.3. Lan truyền ngược theo thời gian.	7
2.1.3. Các hàm kích hoạt thường dùng.....	8
2.2. Mạng nơ ron tích chập.	8
2.2.1. Tổng quan.	8
2.2.2. Lớp tích chập(CONV).	9
2.2.2.1. Tích chập trên bức ảnh dạng 2 chiều (dạng gray).	9
2.2.3. Lớp Pooling (POOL).	16
2.2.3. Lớp Fully Connected (FC).	18
2.2.5. Các siêu tham số của mạng nơ ron tích chập.	18
2.8.6. Mạng LeNet-5 và bài toán nhận diện ký tự số.	23
CHƯƠNG 3: XÂY DỰNG ỨNG DỤNG VÀ KẾT QUẢ THỰC NGHIỆM	40
CHƯƠNG 4: ĐỊNH HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI.....	41

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

1.1. Tên đề tài

- Đề tài: ỨNG DỤNG HỌC SÂU VÀO TRUY VẤN SỰ KIỆN TRONG VIDEO.

1.2. Giới thiệu đề tài

- Trong bối cảnh bùng nổ của công nghệ số, video đã trở thành nguồn thông tin quan trọng và phong phú. Tuy nhiên, việc tìm kiếm và trích xuất thông tin từ khối lượng video khổng lồ vẫn là một thách thức lớn đối với người dùng và các nhà nghiên cứu.
- Học sâu (Deep Learning) với những tiến bộ vượt bậc trong trí tuệ nhân tạo, đã mở ra những giải pháp đột phá cho việc phân tích và truy vấn sự kiện trong video. Các mô hình mạng nơ-ron như CNN, RNN và Transformer có khả năng xử lý đa chiều, giúp nhận diện và kết nối các sự kiện một cách chính xác và nhanh chóng.
- Đề tài nghiên cứu tập trung vào việc:
 - o Khảo sát các phương pháp học sâu tiên tiến
 - o Phát triển mô hình truy vấn sự kiện hiệu quả
 - o Nâng cao khả năng trích xuất thông tin từ video
- Ý nghĩa của nghiên cứu không chỉ dừng lại ở mặt học thuật mà còn mang lại giá trị thực tiễn cao trong nhiều lĩnh vực như an ninh, truyền thông, giáo dục và giải trí.
- Thông qua việc khai thác tiềm năng của học sâu, chúng tôi hướng đến việc cung cấp một công cụ truy vấn video thông minh, chính xác và thân thiện với người dùng.

1.3. Phạm vi nghiên cứu.

Trong khuôn khổ nghiên cứu này, nhóm sinh viên chúng tôi sẽ tập trung vào việc ứng dụng các công nghệ học sâu để phát triển và cải thiện khả năng truy vấn sự kiện trong video. Cụ thể, phạm vi nghiên cứu sẽ bao gồm các nội dung và công nghệ sau:

- Khám Phá và Phát Triển Mô Hình RNN:
 - o Sử dụng mạng nơ-ron hồi tiếp (RNN) để phân tích và nhận diện các chuỗi sự kiện trong video, nhận diện mối quan hệ thời gian giữa các sự kiện diễn ra.
 - o Nghiên cứu cách tối ưu hóa khả năng nhận diện và phân loại các sự kiện trong video bằng cách áp dụng các kỹ thuật như LSTM (Long Short-Term Memory) và GRU (Gated Recurrent Units).
- Khảo Sát Mô Hình Zorro-shot:

- Nghiên cứu mô hình Zorro-shot trong việc cải tiến độ chính xác của việc truy vấn sự kiện và tăng khả năng nhận diện đối tượng trong video.
- Tích hợp Zorro-shot vào quy trình phân tích để nâng cao hiệu quả trích xuất thông tin.
- CNN:
 - Được sử dụng để trích xuất và mã hóa các đặc trưng hình ảnh thành vector embedding thông qua mô hình CLIP của OpenAI, cho phép tìm kiếm và kết nối hình ảnh dựa trên truy vấn văn bản một cách hiệu quả.
- Tích Hợp OpenAI CLIP:
 - Khai thác sức mạnh của mô hình OpenAI CLIP trong việc phân tích ngữ cảnh và nội dung hình ảnh trong video, cho phép liên kết giữa hình ảnh và văn bản để truy vấn thông tin cụ thể.
 - Nghiên cứu cách CLIP có thể cải thiện khả năng nhận diện các chủ đề và sự kiện trong video thông qua các mô tả văn bản.
- Ứng Dụng ANN (Mạng Nơ-ron Tự):
 - Sử dụng các mạng nơ-ron tự (Feedforward Neural Networks) để phân tích dữ liệu đầu vào từ video, từ đó trích xuất các đặc trưng quan trọng giúp nhận diện sự kiện.
 - Thực hiện các thí nghiệm để đánh giá hiệu quả của ANN trong việc phân loại và dự đoán sự kiện.
- Sử Dụng ORC (Optical Character Recognition):
 - Áp dụng công nghệ nhận diện ký tự quang học (ORC) để trích xuất thông tin từ văn bản trong video, hỗ trợ trong việc truy vấn sự kiện một cách chính xác hơn.
 - Nghiên cứu cách tích hợp ORC vào quy trình phân tích video để cải thiện khả năng dịch ngữ cảnh của sự kiện.

1.4. Ý nghĩa thực tiễn của đề tài:

- Ý nghĩa đối với công nghệ và xã hội
 - Video là nguồn dữ liệu ngày càng phong phú, nhưng việc tìm kiếm thông tin trong các tệp video lớn vẫn là một thách thức. Đề tài ứng dụng học sâu vào truy vấn sự kiện trong video nhằm tối ưu hóa quá trình trích xuất thông tin, giảm thời gian xử lý và tăng độ chính xác. Kết quả nghiên cứu mang lại giá trị thực tiễn cao trong các lĩnh vực như:
- An ninh: Hệ thống giám sát tự động nhận diện sự kiện bất thường.
 - Giáo dục và truyền thông: Tìm kiếm và tóm tắt nội dung hữu ích từ các bài giảng hoặc chương trình video.
 - Giải trí: Gợi ý nội dung phù hợp hoặc phân tích video theo sở thích cá nhân.
- Đóng góp của nhóm sinh viên thực hiện

- Phát triển chuyên môn: Nhóm nghiên cứu các kỹ thuật học sâu hiện đại như RNN, CLIP (OpenAI), và Zero-shot Learning.
- Ứng dụng thực tế: Phát triển công cụ truy vấn thông minh, giúp cải thiện trải nghiệm người dùng khi làm việc với dữ liệu video.

1.7. Các mô hình, công cụ được sử dụng trong đề tài:

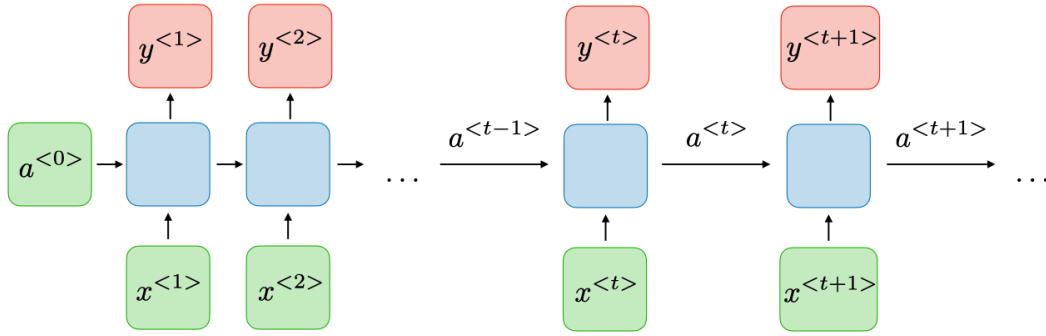
- CLIP (OpenAI): Kết nối hình ảnh và ngôn ngữ tự nhiên, giúp hiểu ngữ cảnh sự kiện.
- RNN: Phân tích chuỗi sự kiện có mối quan hệ thời gian.
- Zero-shot Learning: Phân loại sự kiện mà không cần dữ liệu huấn luyện trước.
- OCR: Trích xuất văn bản từ video, hỗ trợ truy vấn nội dung cụ thể.
- gTTS: Chuyển văn bản tóm tắt sự kiện thành giọng nói.
- CNN: trong mô hình CLIP giúp chuyển đổi hình ảnh thành vector đặc trưng để tìm kiếm và so khớp với văn bản.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Mạng neural hồi quy.

2.1.1. Kiến trúc của một mạng RNN truyền thống

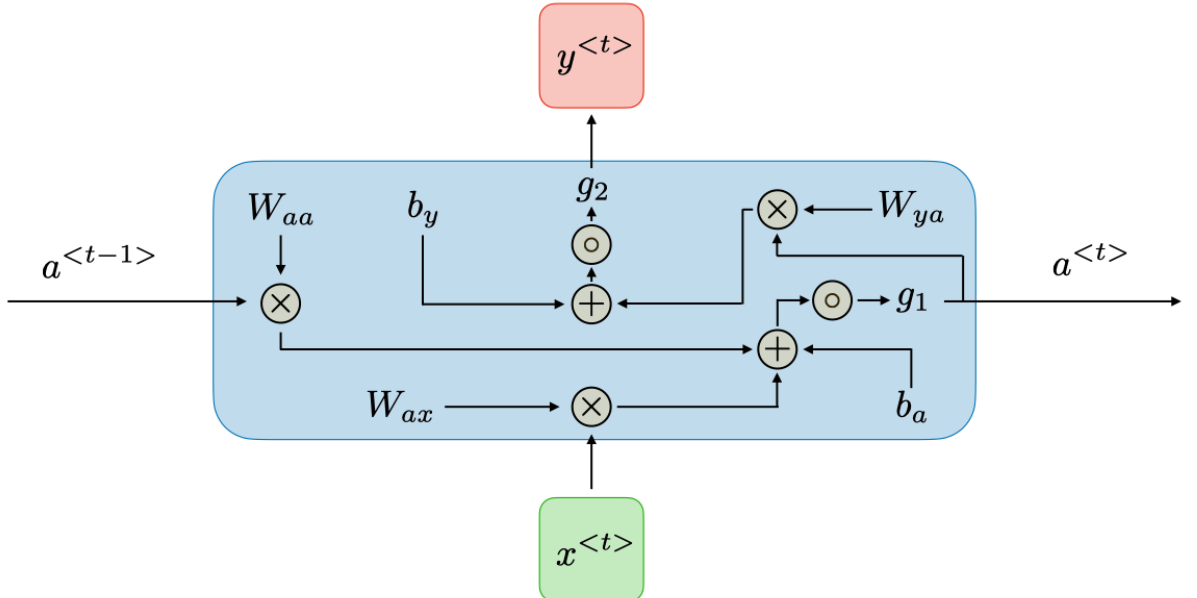
- Các mạng neural hồi quy, còn được biến đổi như là RNNs, là một lớp của mạng neural cho phép đầu ra được sử dụng như đầu vào trong khi có các trạng thái ẩn. Thông thường là như sau:



Tại mỗi bước t , giá trị kích hoạt $a^{<t>}$ và đầu ra $y^{<t>}$ được biểu diễn như sau:

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \quad \text{và} \quad y^t = g_2(W_{ya}a^{<t>} + b_y)$$

Với $W_{ax}, W_{aa}, W_{ya}, b_a, b_u$ là các hệ số được chia sẻ tạm thời và g_1, g_2 là hàm kích hoạt.

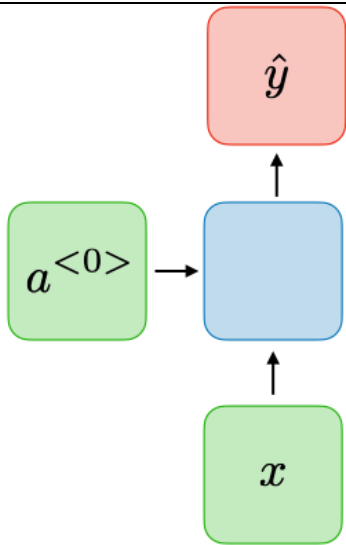


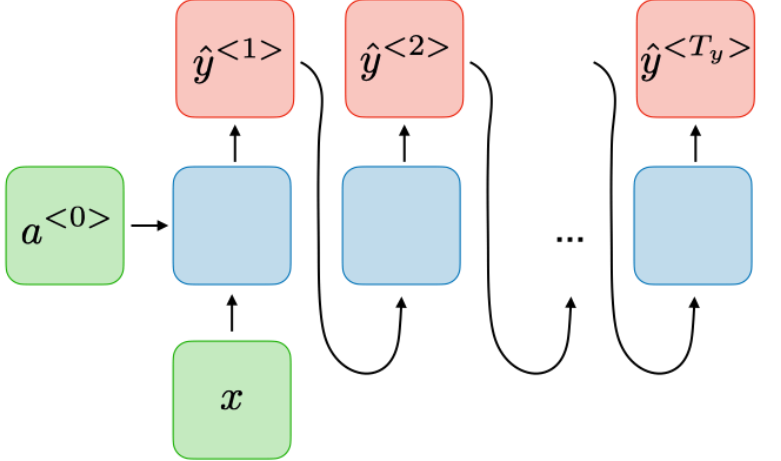
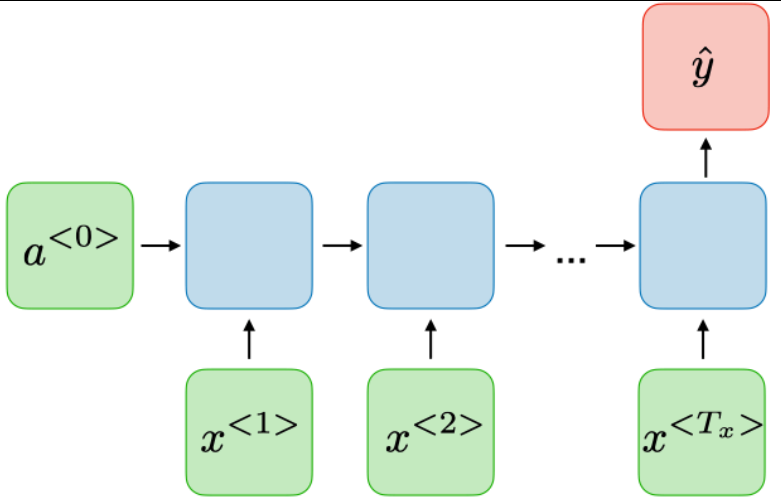
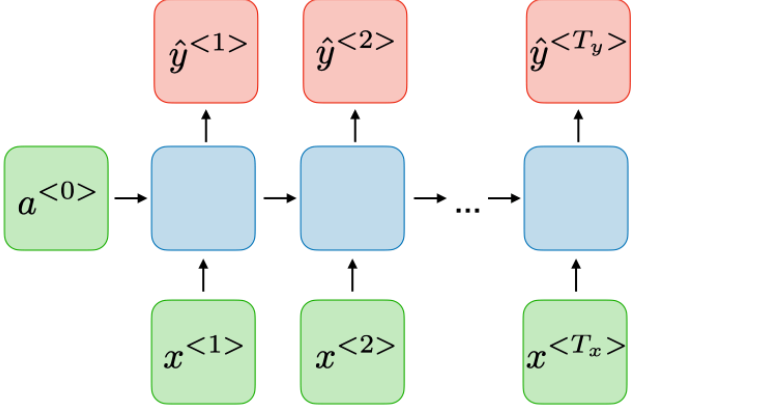
- Ưu và nhược điểm của một kiến trúc RNN thông thường được tổng kết ở bảng dưới đây:

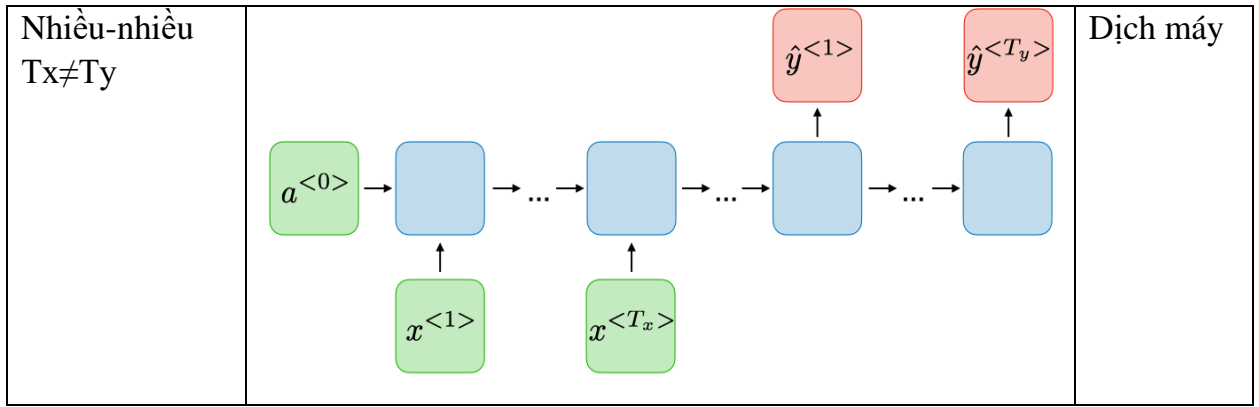
Ưu điểm	Hạn chế
<ul style="list-style-type: none"> • Khả năng xử lý đầu vào với bất kì độ dài nào • Kích cỡ mô hình không tăng theo kích cỡ đầu vào • Quá trình tính toán sử dụng các thông tin cũ • Trọng số được chia sẻ trong suốt thời gian 	<ul style="list-style-type: none"> • Tính toán chậm • Khó để truy cập các thông tin từ một khoảng thời gian dài trước đây • Không thể xem xét bất kì đầu vào sau này nào cho trạng thái hiện tại

2.1.2. Ứng dụng của RNNs.

- Các mô hình RNN hầu như được sử dụng trong lĩnh vực xử lý ngôn ngữ tự nhiên và ghi nhận tiếng nói. Các ứng dụng khác được tổng kết trong bảng dưới đây:

Các loại RNN	Hình minh hoạ	Ví dụ
Một-Một $T_x=T_y=1$		Mạng neural truyền thống

<p>Một-nhiều $T_x=1, T_y>1$</p>		<p>Sinh nhạc</p>
<p>Nhiều-một $T_x>1, T_y=1$</p>		<p>Phân loại ý kiến</p>
<p>Nhiều-nhiều $T_x=T_y$</p>		<p>Ghi nhận thực thể tên</p>



2.1.3. Hàm mất mát.

Trong trường hợp của mạng neural hồi quy, hàm mất mát L của tất cả các bước thời gian được định nghĩa dựa theo mất mát ở mọi thời điểm như sau:

$$L(\hat{y}, y) = \sum_{t=1}^T L(\hat{y}^{<t>}, y^t)$$

Giải thích chi tiết công thức:

1. \hat{y} : Đây là đầu ra dự đoán của mô hình tại mỗi bước thời gian t . Nó được tính toán dựa trên đầu vào tại thời điểm đó và trạng thái ẩn từ các bước trước.
2. y : Đây là giá trị thực tế (ground truth) tại mỗi bước thời gian t , được sử dụng để so sánh với giá trị dự đoán \hat{y} .
3. $L(\hat{y}^t, y^t)$: Đây là hàm mất mát tại thời điểm t , đo lường sự khác biệt giữa giá trị dự đoán \hat{y}^t và giá trị thực tế y^t . Hàm mất mát này có thể là:

- MSE (Mean Squared Error): Thường dùng cho bài toán hồi quy.

$$L(\hat{y}^t, y^t) = (\hat{y}^t - y^t)^2$$

- Cross-Entropy Loss: Thường dùng cho bài toán phân loại.

$$L(\hat{y}^t, y^t) = - \sum_i y_i^t \log \hat{y}_i^t$$

4. $\sum_{t=1}^T$: Tổng hợp mất mát trên toàn bộ chuỗi thời gian từ bước $t=1$ đến $t=T$, nơi T là độ dài của chuỗi.

2.1.3. Lan truyền ngược theo thời gian.

Lan truyền ngược được hoàn thành ở mỗi một thời điểm cụ thể. Ở bước T , đạo hàm của hàm mất mát L với ma trận trọng số W được biểu diễn như sau:

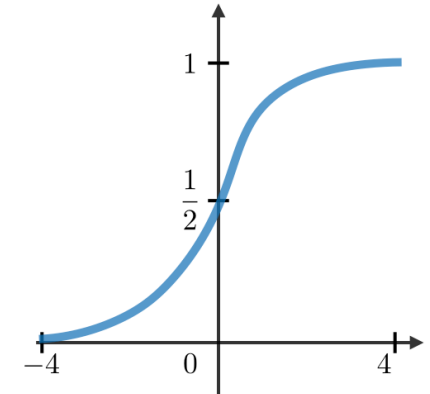
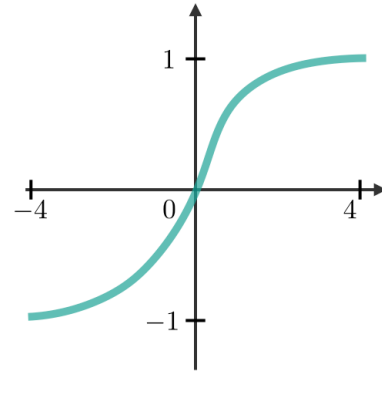
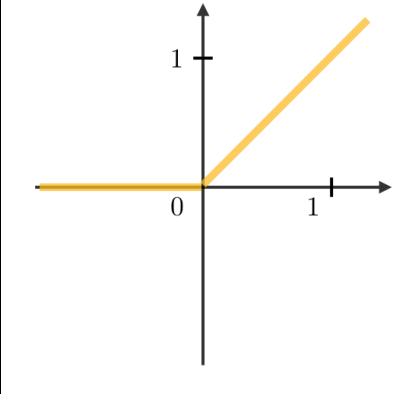
$$\frac{\partial L^{(T)}}{\partial W} = \sum_{t=1}^T \frac{\partial L^{(t)}}{\partial W} \bigg|_t$$

Giải thích công thức:

- $\frac{\partial L^{(T)}}{\partial W}$: Đây là đạo hàm của hàm mất mát tổng cộng L (tính trên toàn bộ chuỗi thời gian từ $t=1$ đến $t=T$) đối với ma trận trọng số W . Mục tiêu là sử dụng giá trị này để cập nhật trọng số W nhằm giảm thiểu hàm mất mát.
- $\sum_{t=1}^T$: Tổng hợp các đạo hàm tại từng bước thời gian t . Điều này phản ánh rằng lan truyền ngược trong RNN phải xử lý toàn bộ chuỗi thời gian, thay vì chỉ một bước duy nhất.
- $\frac{\partial L^{(T)}}{\partial W}$: Đây là đạo hàm của hàm mất mát tại thời điểm t đối với trọng số W . Nó được tính toán dựa trên trạng thái ẩn và đầu ra tại thời điểm đó.

2.1.3. Các hàm kích hoạt thường dùng.

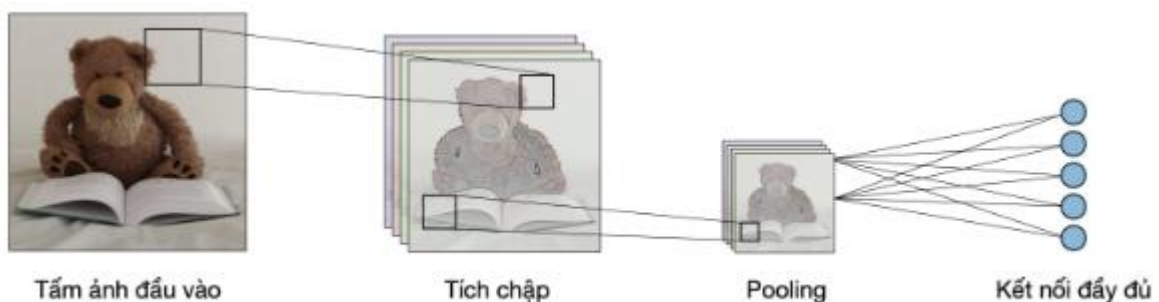
Các hàm kích hoạt thường dùng trong các modules RNN được miêu tả như sau:

Sigmoid	Tanh	RELU
$g(z) = \frac{1}{1 + e^{-z}}$	$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0, z)$
		

2.2. Mạng nơ ron tích chập.

2.2.1. Tổng quan.

Kiến trúc truyền thống của một mạng CNN- Mạng nơ ron tích chập (Convolutional neural networks) là một kiến trúc mạng nơ-ron đặc biệt được thiết kế để xử lý dữ liệu không gian như ảnh và âm thanh. Còn được biết đến với tên CNNs, là một dạng mạng neural được cấu thành bởi các tầng sau:

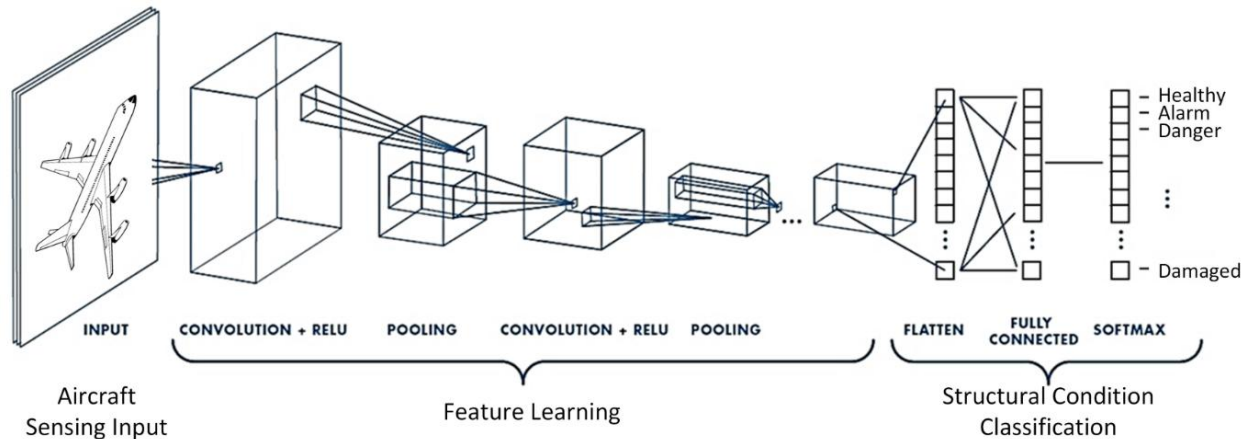


Hình 2.8.1.1: ví dụ minh họa.

Tầng tích chập và tầng pooling có thể được hiệu chỉnh theo các siêu tham số (hyperparameters).

Có 3 kiểu lớp:

- Lớp tích chập (CONV)
- Lớp Pooling (POOL)
- Lớp Fully Connected (FC)



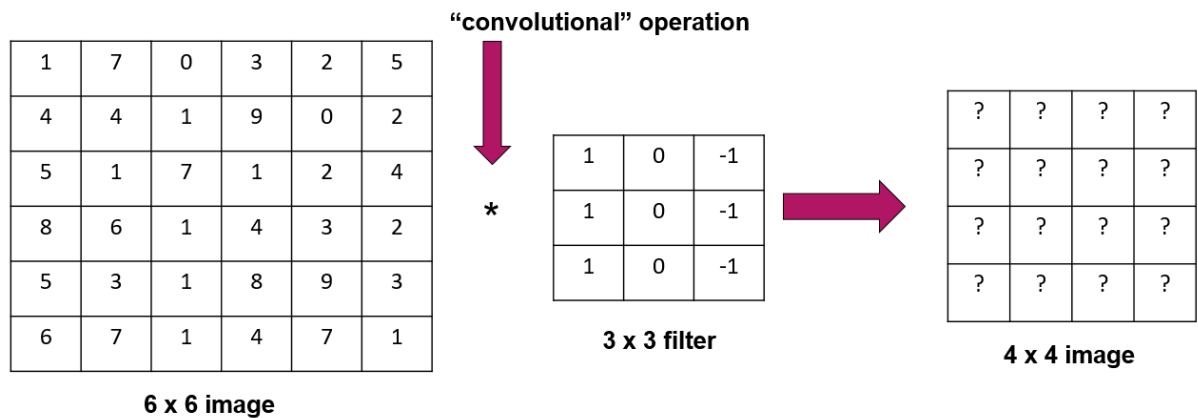
Hình 2.8.1.2: Mạng tích chập cơ bản.

Kiến trúc mạng nơ ron tích chập nhận đầu vào là một bức ảnh và đầu ra được minh họa thể hiện nguyên lý hoạt động của hệ thống, có 2 quá trình đó là Feature Learning và Structural Condition Classification.

- Feature Learning (học đặc trưng) là quá trình mạng học các đặc trưng (features) từ dữ liệu đầu vào. Các lớp tích chập (convolutional layers) và lớp gộp (pooling layers) được sử dụng để trích xuất các đặc trưng cấp cao hơn (như hình dạng, khuôn mẫu) ở các lớp tiếp theo. Quá trình này giúp mạng học được các biểu diễn (representations) mang ý nghĩa của dữ liệu, thay vì chỉ lưu trữ các giá trị pixel thô. Việc học các đặc trưng phù hợp là rất quan trọng để giúp mạng có thể phân loại và nhận dạng dữ liệu một cách chính xác.
- Structural Condition Classification (phân loại điều kiện cấu trúc) là quá trình sau khi đã học được các đặc trưng từ dữ liệu đầu vào, giai đoạn này chúng sẽ được đưa vào các lớp kết nối đầy đủ (fully connected layers) để thực hiện quá trình phân loại các điều kiện cấu trúc hoặc các đối tượng trong dữ liệu. Các lớp kết nối đầy đủ (fully connected layers) được sử dụng ở giai đoạn này để biến đổi các đặc trưng được học thành các nhãn lớp (class labels). Lớp đầu ra sẽ dự đoán lớp hoặc trạng thái của điều kiện cấu trúc.

2.2.2.Lớp tích chập(CONV).

2.2.2.1.Tích chập trên bức ảnh dạng 2 chiều (dạng gray).

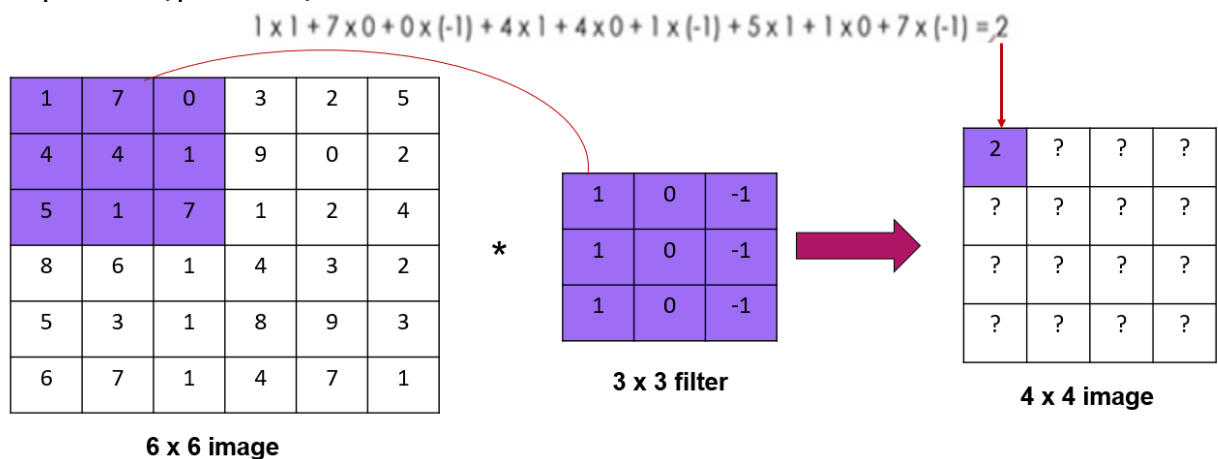


Hình 2.8.2.1.1: 2 chiều (dạng gray).

Đầu tiên xét bức ảnh 6x6 , bức ảnh dạng gray kích thước 6x6 chúng ta sẽ thử thực 1 phép toán tích chập bức ảnh của chúng ta với 1 cái Filter (hay còn gọi là bộ lọc)

Có 2 toán tử là 2 cái ma trận, toán tử 1 là bức ảnh còn toán tử 2 là filter , phép toán tích chập được ký hiệu * , output 4x4

B1: lấy 1 ma trận có kích thước 3x3 trong bức ảnh 6x6 bằng kích thước filter và thực hiện phép tích chập 2 ma trận vs nhau.



Hình 2.8.2.1.2: ma trận có kích thước 3x3 trong bức ảnh 6x6.

Ta có : $1 \times 1 + 7 \times 0 + 0 \times (-1) + 4 \times 1 + 4 \times 0 + 1 \times (-1) + 5 \times 1 + 1 \times 0 + 7 \times (-1) = 2$

B2: chúng ta di chuyển ma trận ảnh 3x3 sang trái 1 vị trí và cứ thế nếu hết hàng thì sẽ di chuyển xuống dưới 1 vị trí và lặp lại B1.

$$7 \times 1 + 0 \times 0 + 3 \times (-1) + 4 \times 1 + 1 \times 0 + 9 \times (-1) + 1 \times 1 + 7 \times 0 + 1 \times (-1) = -1$$

1	7	0	3	2	5
4	4	1	9	0	2
5	1	7	1	2	4
8	6	1	4	3	2
5	3	1	8	9	3
6	7	1	4	7	1

6 x 6 image

*

1	0	-1
1	0	-1
1	0	-1

3 x 3 filter



2	-1	?	?
?	?	?	?
?	?	?	?
?	?	?	?

4 x 4 image

1	7	0	3	2	5
4	4	1	9	0	2
5	1	7	1	2	4
8	6	1	4	3	2
5	3	1	8	9	3
6	7	1	4	7	1

6 x 6 image

*

1	0	-1
1	0	-1
1	0	-1

3 x 3 filter



2	-1	4	2
8	-3	4	6
9	-3	-5	4
16	0	-16	10

4 x 4 image

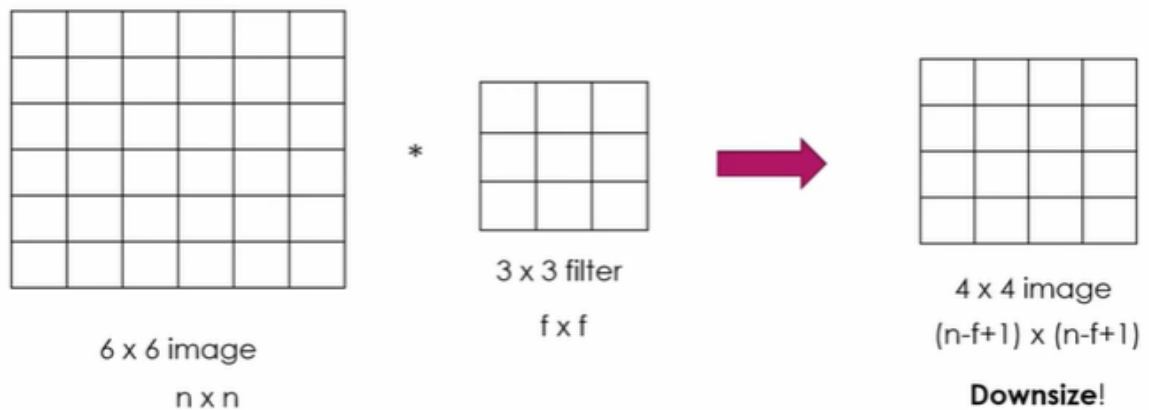
Hình 2.8.2.1.3: di chuyển ma trận ảnh 3x3.

Ta nhận thấy kích thước 4x4 được quy từ việc dịch chuyển ma trận có kích thước 3x3 trong bức ảnh trên mỗi hàng thì có 4 vị trí và mỗi cột cũng có 4 vị trí cho nên output của tích chập 2 ma trận này là 1 ma trận có kích thước 4x4 $\Rightarrow 6-3+1=4$ (kích thước).

Các thư viện được sử dụng trong code:

Python: conv_forward
Tensorflow: tf.nn.conv2d
Keras: Conv2D

- Python: conv_forward
- Tensorflow: tf.nn.conv2d
- Keras: Conv2D

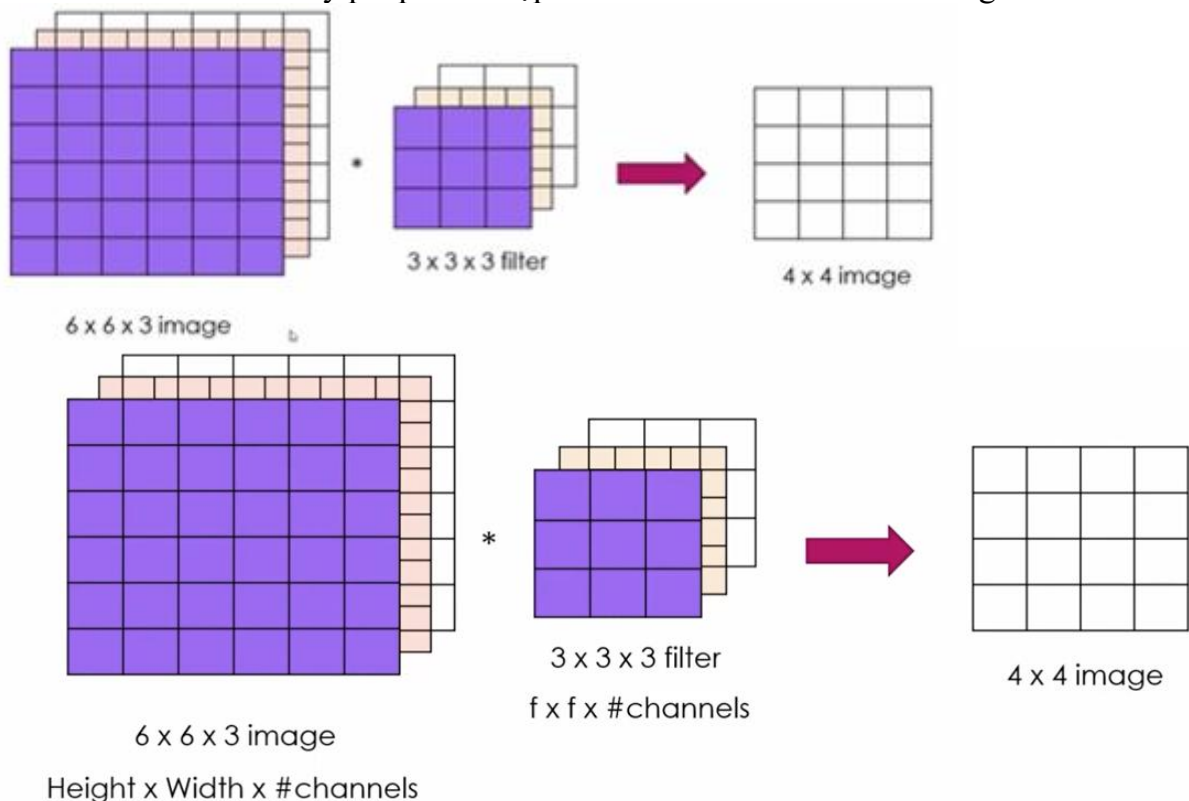


Kết luận : Chúng ta có được công thức tổng quát để tìm được kích thước ảnh Output
 $\text{Size} = (n-f+1) \times (n-f+1)$

2.2.2.2. Tích chập trên bức ảnh dạng 3 chiều RGB.

Đi vào bài toán thực tế hơn, bởi hiện nay tất cả những bức ảnh chúng ta chụp ha được nhìn thấy đều là dạng ảnh RGB với 3 kênh màu chủ đạo Red, Green, Blue hay còn gọi là RGB.

Phép tích chập cho bức ảnh có chiều sâu (RGB): chúng ta sẽ có được filter cũng có 3 lớp như ảnh đầu vào để xử lý phép tích chập cho bức ảnh đầu vào của chúng ta



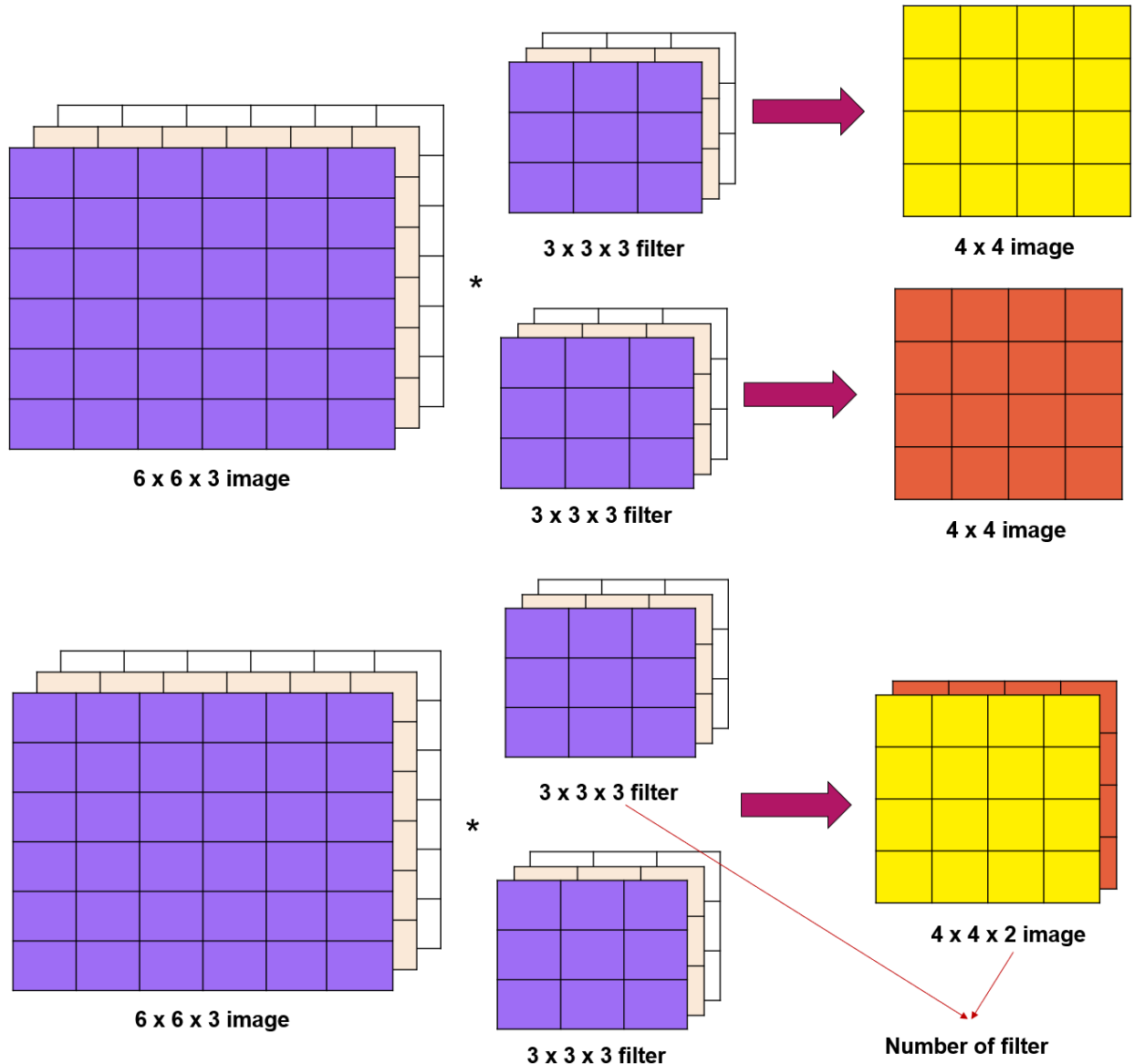
Hình 2.8.2.2.1: xử lý phép tích chập cho bức ảnh đầu vào.

Số channel ở đây không nhất thiết sẽ là = 3 mà nó còn có thể tăng lên trong quá trình huấn luyện trong các layer sâu hơn và bất di bất dịch số channel của ảnh thay đổi thì channel của filter cũng thay đổi theo

Lưu ý : chúng ta thực hiện phép tích chập trên không gian 3 chiều nhưng đầu ra nó vẫn chỉ là bức ảnh trên không gian 2 chiều.

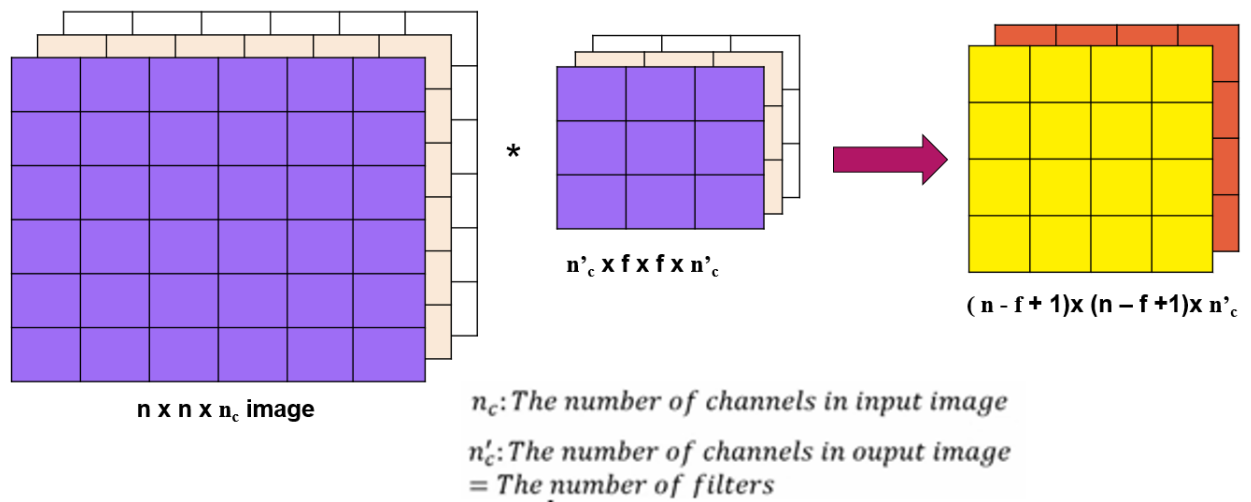
Vì sao Output lại không có chiều sâu? : chúng ta có quy tắc phép tích chập trong không gian 3 chiều là chúng ta chỉ dịch chuyển theo chiều ngang và dọc và không dịch chuyển theo chiều sâu.

Phép tích chập được thực hiện giữa từng cặp kênh tương ứng của ảnh đầu vào và bộ lọc. Kết quả của 3 phép tích chập này được cộng lại để tạo thành 1 giá trị đầu ra. Nhận thấy rằng lần tích chập tích theo thì sẽ không còn là bức ảnh 3 chiều như đầu vào thì chúng ta sẽ xử lý bằng cách thêm số lượng filter để tạo ra các chiều mới .



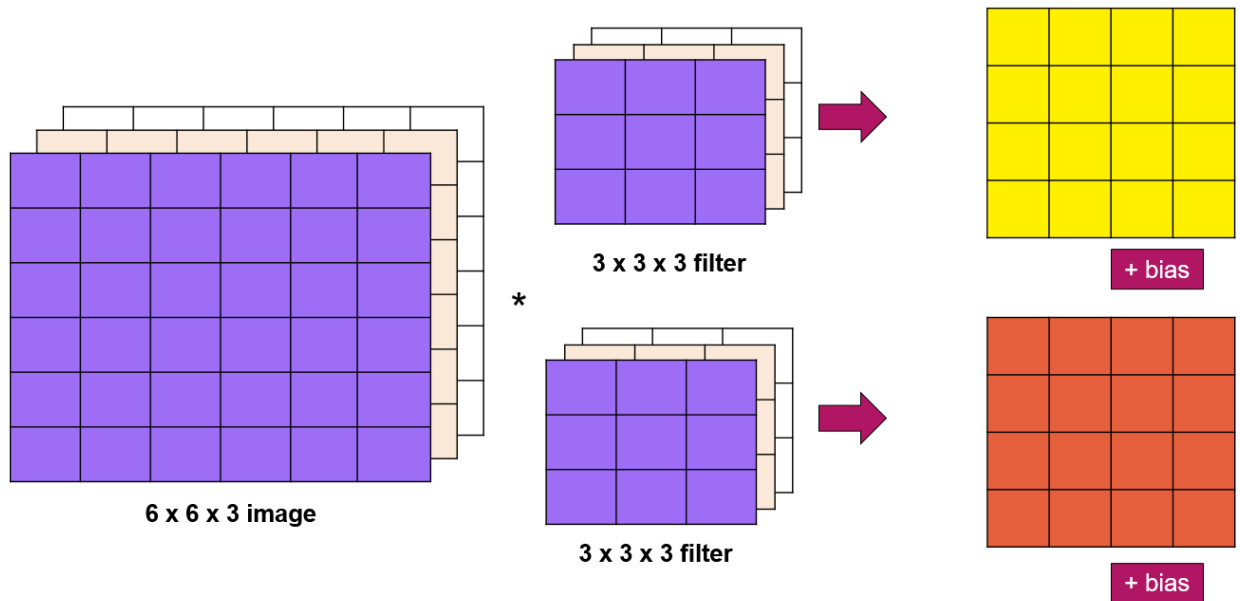
Hình 2.8.2.2.2: tạo ra các chiều mới.

Một bức ảnh của 1 vật thể có rất nhiều góc cạnh khác nhau có rất nhiều đường nét đặc trưng khác nhau cho nên chúng ta phải tăng số lượng filter để có thể phát hiện ra nhiều góc cạnh đặc trưng khác nhau trong bức ảnh Input và mỗi filter đây có các giá trị khác nhau.



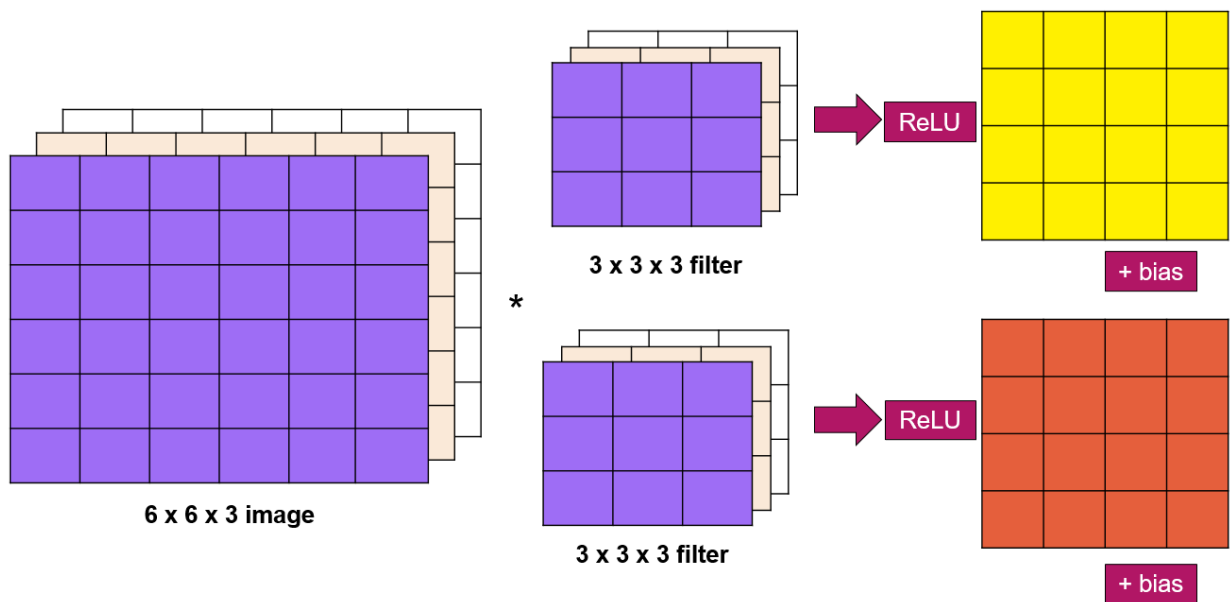
Hình 2.8.2.2.3: góc cạnh đặc trưng khác nhau trong bức ảnh Input và mỗi filter.

Trong đó : n_c : số kênh của ảnh đầu vào (Input)
 n'_c : số kênh của ảnh đầu ra (Output) = số lượng filter
 trong 1 layer của tích chập:



Hình 2.8.2.2.4: số kênh của ảnh đầu vào.

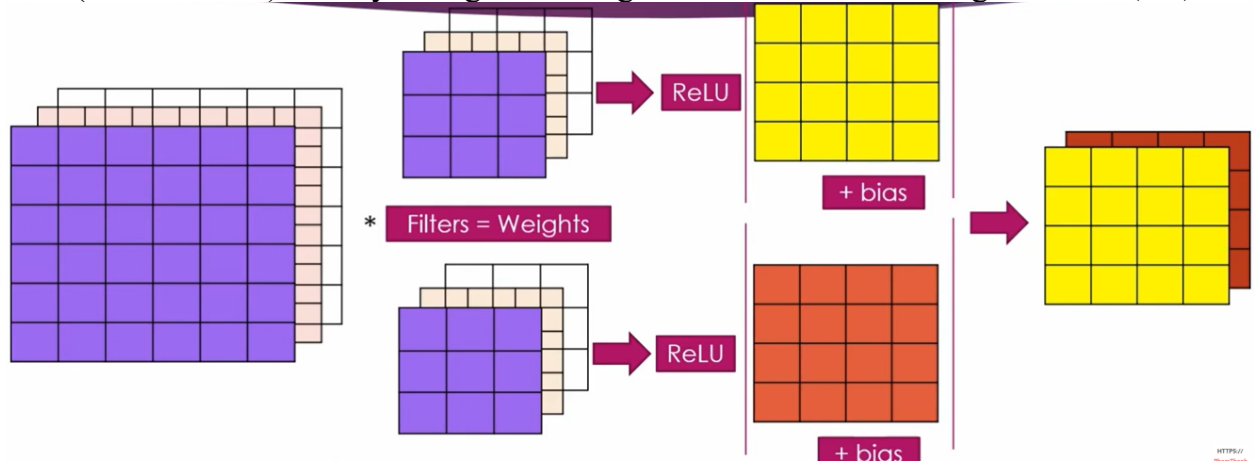
Chúng ta cộng từng giá trị trong ma trận ảnh output với một số thực Bias



Hình 2.8.2.2.5: ma trận ảnh output với một số thực Bias.

Sau khi cộng Bias thì chúng ta sẽ đưa vào hàm kích hoạt , đơn giản chúng ta chỉ việc đưa từng giá trị có trong ma trận ảnh vào hàm kích hoạt và có được giá trị tương ứng cho ô đấy.

* Lưu ý: Bởi phép tích chập là phép toán tuyến tính và trong bài toán thực tế thì có các đặc trưng phi tuyến và để tránh các vấn đề như Overfitting thì chúng ta phải cần hàm kích hoạt (Act Function) . Ở đây chúng ta sử dụng hàm ReLU được định nghĩa $= \max(0, x)$.

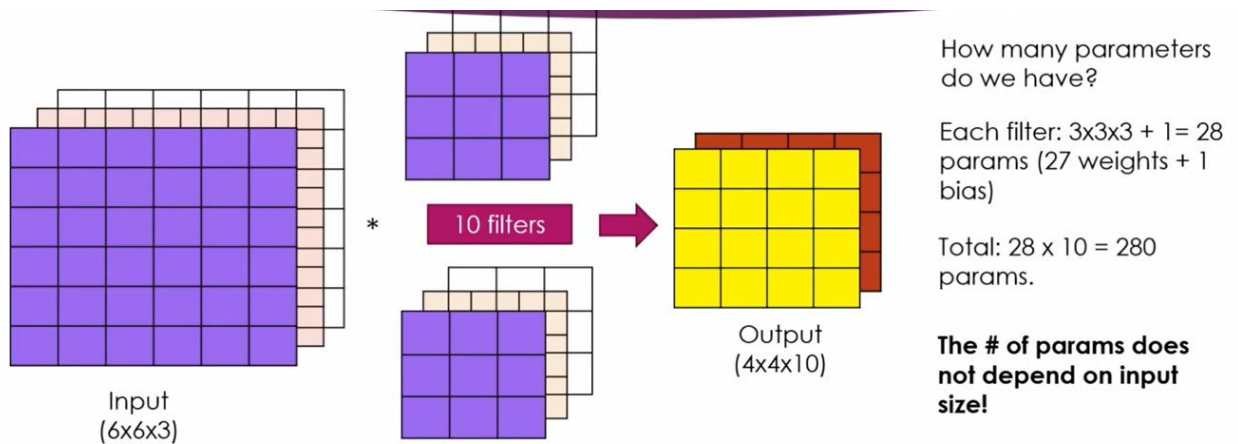


Hình 2.8.2.2.6: dụng hàm ReLU.

- Nhận xét: một mạng nơ ron đơn thuần so tích hợp nhiều layer liên tiếp nhau trong đó Input layer này là Output của layer trước nó. Cũng có thể trong 1 mạng nơ ron tích chập thì nó không phải chỉ chứa các layer tích chập mà nó còn có một số layer như là Pooling , Fully Connected.

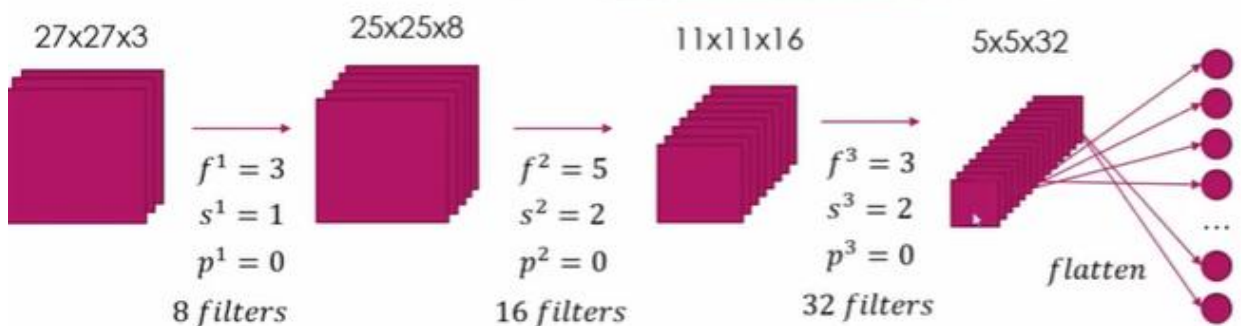
Cách tính số lượng tham số có trong 1 mạng nơ ron tích chập :

Trong mạng nơ ron tích chập số lượng tham số chỉ dựa vào kích thước và số lượng của Filter cho nên dù Input đầu vào là bức ảnh có hàng triệu pixel thì số lượng tham số cũng không thay đổi.



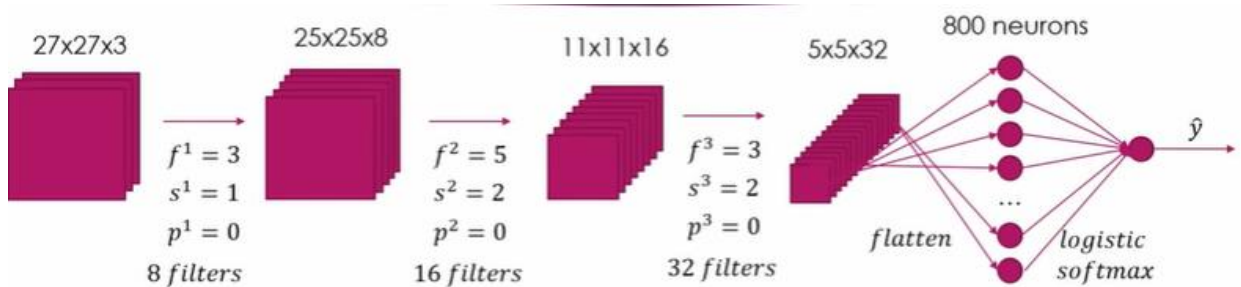
Hình 2.8.2.2.7: mạng nơ ron tích chập số lượng tham.

Bài toán ví dụ: cho bức ảnh RGB ban đầu có kích thước 27x27x3



Hình 2.8.2.2.8: Bài toán ví dụ.

Sau quá trình qua nhiều layer tích chập với các giá trị thông số kích thước filter, stride khác nhau. Chúng ta sẽ dãn phẳng 5x5x32 (800 pixel) thành 800 entry vectơ hay còn gọi là 800 neuron

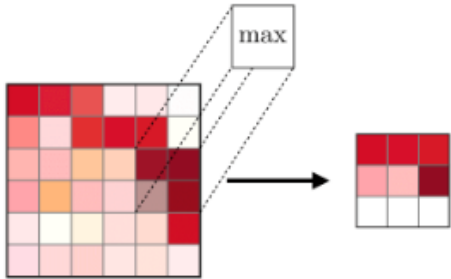
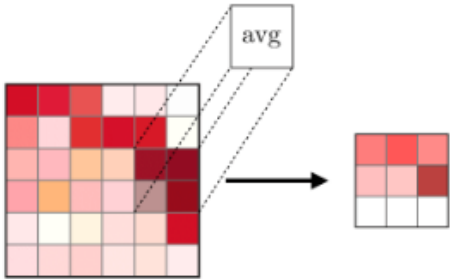


Hình 2.8.2.2.9: trị thông số kích thước filter.

Và chúng ta sẽ cho 800 neuron kết nối tới 1 neuron và đưa ra Output. Đây là 1 thiết kế mạng nơ ron tích chập đơn giản giúp chúng ta giải quyết bài toán nhận dạng 1 bức ảnh, 1 con số .

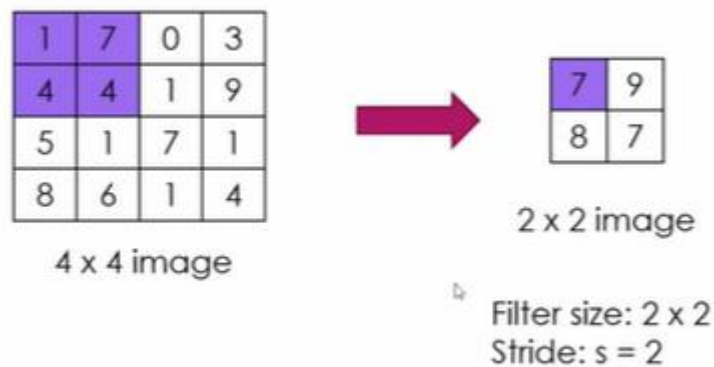
2.2.3.Lớp Pooling (POOL).

Pooling có 2 kiểu: Max pooling và Average pooling

Kiểu	Max pooling	Average pooling
Chức năng	Từng phép pooling chọn giá trị lớn nhất trong khu vực mà nó đang được áp dụng	Từng phép pooling tính trung bình các giá trị trong khu vực mà nó đang được áp dụng
Minh họa		
Nhận xét	<ul style="list-style-type: none"> • Bảo toàn các đặc trưng đã phát hiện • Được sử dụng thường xuyên 	<ul style="list-style-type: none"> • Giảm kích thước feature map • Được sử dụng trong mạng LeNet

2.2.3.1. Max pooling.

- Kiểu Max pooling: Ví dụ:



Hình 2.8.3.1: Max pooling.

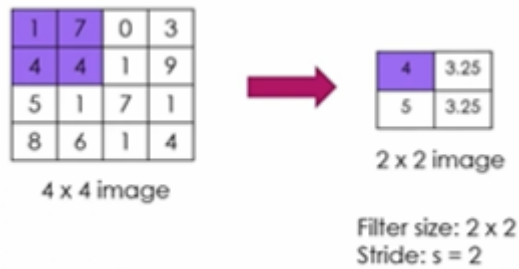
Nhận xét: đơn giản chúng ta chọn con số lớn nhất trong ma trận đang tham chiếu tới.

Quy tắc: Max pooling sẽ xét lần lượt từng channel một. Khác so với Conv là Conv thì sẽ thực hiện tích chập trên tất cả channel cùng 1 lúc, còn đối với max pooling chúng ta sẽ thực hiện max pooling trên từng channel một.

Mỗi bước max pooling chúng ta chỉ có duy nhất filter chứ ko có nhiều filter tương ứng và số channel của Input và Output hoàn toàn bằng nhau.

2.2.3.2. Average pooling.

- Kiểu Average pooling : chúng ta tính trung bình các giá trị được tham chiếu tới .
Ví dụ:



Hình 2.8.3.2: Average pooling.

2.2.3.3. Đánh giá.

- Max pooling thường được sử dụng nhiều hơn so với Average pooling vì thường những đặc trưng dễ nhất, nổi bật nhất nó nằm ở pixel có giá trị lớn nhất. thay vì dùng Average pooling nó triệt tiêu lẫn nhau nó làm mất đi feature của bức ảnh thì max pooling nó sẽ chú trọng vào những đặc trưng nổi bật nhất.
- Pooling thường ta người ta sử dụng để giảm cái size của bức ảnh xuống và trích xuất những đặc trưng quan trọng từ dữ liệu cho nên padding cũng rất ít được sử dụng trong pooling vì padding là muốn giữ nguyên kích thước của ảnh ban đầu nó trái ngược so với mục đích sử dụng của pooling .
- Kích thước của Output khi sử dụng pooling:

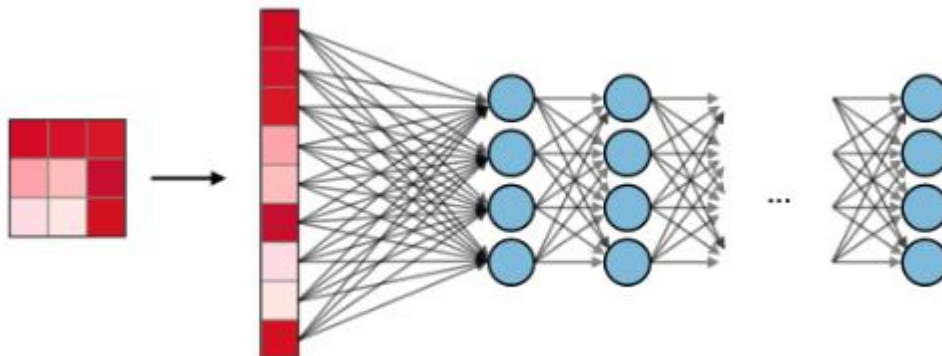
$$\text{Input: } n_H \times n_W \times n_C$$

$$\text{Output: } \left\lfloor \frac{n_H - f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n_W - f}{s} + 1 \right\rfloor \times n_C$$

Và đặc biệt Pooling không có tham số để học bởi vì đơn giản nó không phải nhân hay cộng một giá trị nào ở trong filter cả, nó chỉ lấy giá trị trung bình hoặc lớn nhất trong giá trị được tham chiếu trên Input thôi.

2.2.3.Lớp Fully Connected (FC).

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.



Hình 2.8.3.1:Lớp Fully Connected (FC).

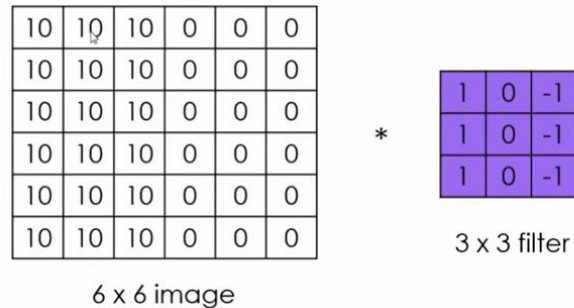
2.2.5.Các siêu tham số của mạng nơ ron tích chập.

Có 3 loại tham số chính: Filter(bộ lọc), Padding, Stride (bước nhảy)

2.2.5.1. Filter (bộ lọc).

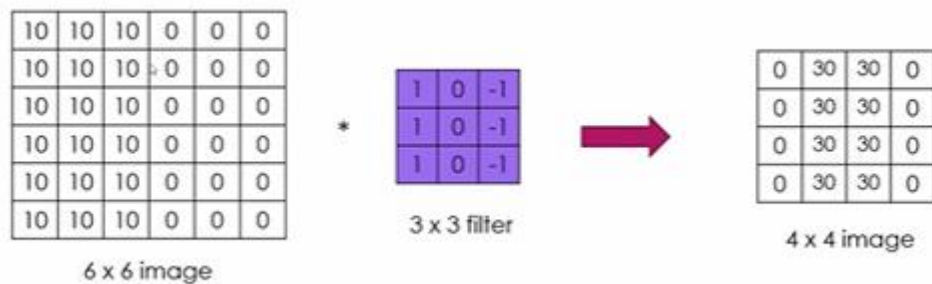
Lớp tích chập là cốt lõi của bất kỳ CNN nào. Nó được sử dụng để trích xuất thông tin từ đầu vào của nó thông qua việc sử dụng một số bộ lọc được tự động dạy để phát hiện các tính năng nhất định trong hình ảnh.

Filter hay còn được gọi là bộ lọc

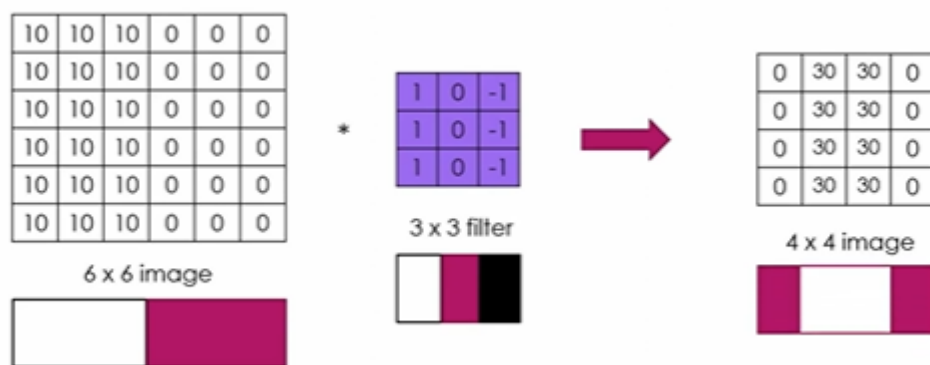


Hình 2.8.5.1.1: Filter (bộ lọc).

Ta có thể thấy để phân biệt hay làm nổi bật giữa màu trắng và màu tím lên thì chúng ta dùng mắt thường sẽ thấy được ngay ranh giới đấy (cột 3 và cột 4). Và ở đây chúng ta cần trang bị cái mắt cho máy tính để máy tính nhận biết được ranh giới đấy bằng Filter thì filter này có cái mắt tương tự như mắt chúng ta, và cách tìm nó như thế nào.



Hình 2.8.5.1.2: cách tính của Filter (bộ lọc).



Hình 2.8.5.1.3: Filter (bộ lọc) trong thực tế.

Nhận xét: chúng ta nhìn ra được cái Filter sẽ làm Highlight (làm đậm) cạnh ngăn cách giữa cột 3 và cột 4 lên.

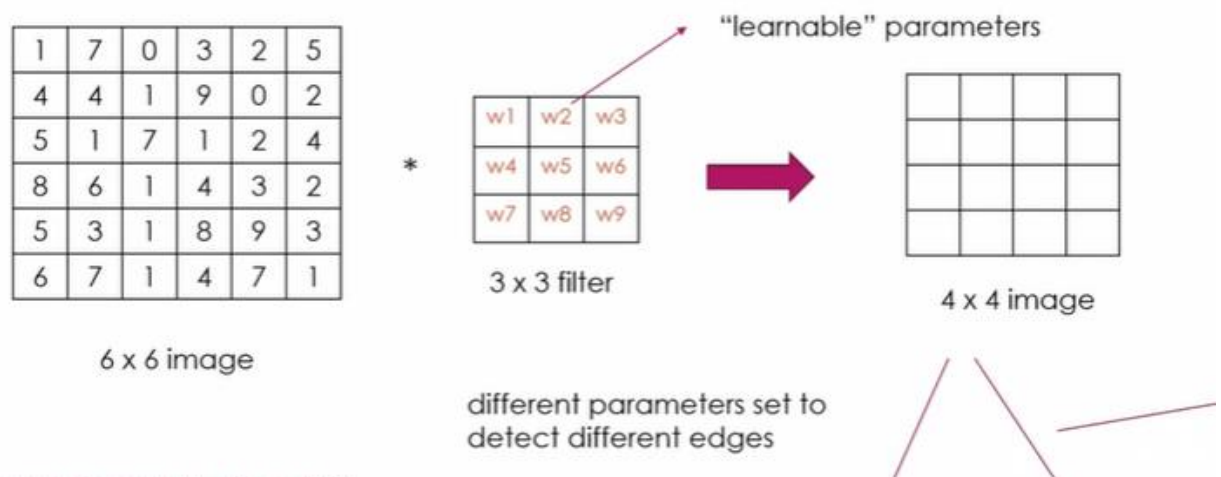
- Filter là một con mắt quét qua toàn bộ bức ảnh gốc để đưa ra 1 bức ảnh mới mà trong đấy những cái cạnh nó được làm nổi bật lên.

- người ta dùng Filter 1,0,-1 này để phát hiện các cạnh thẳng đứng, chúng ta còn có nhiều filter khác có giá trị khác nhau giúp chúng ta phát hiện các cái loại cạnh khác nhau.
- Một số loại Filter: - Sobel , Scharr filter: giúp phát hiện các cạnh theo phương thẳng đứng , nằm ngang nhưng nó có thể phụ vụ cho một số chuyên biệt cụ thể nào đấy, có thể sẽ là highlight đậm hơn một số vị trí

Đối với một số cạnh nghiêng 35, 40 , 45 độ chúng ta không thể tự đi thiết kế từng cái filter để là nhận dạng cho bức ảnh đó cả cho nên chúng ta phải để máy tự học, tự đưa ra các giá trị phù hợp để giải quyết được các bài toán phù hợp khác nhau.



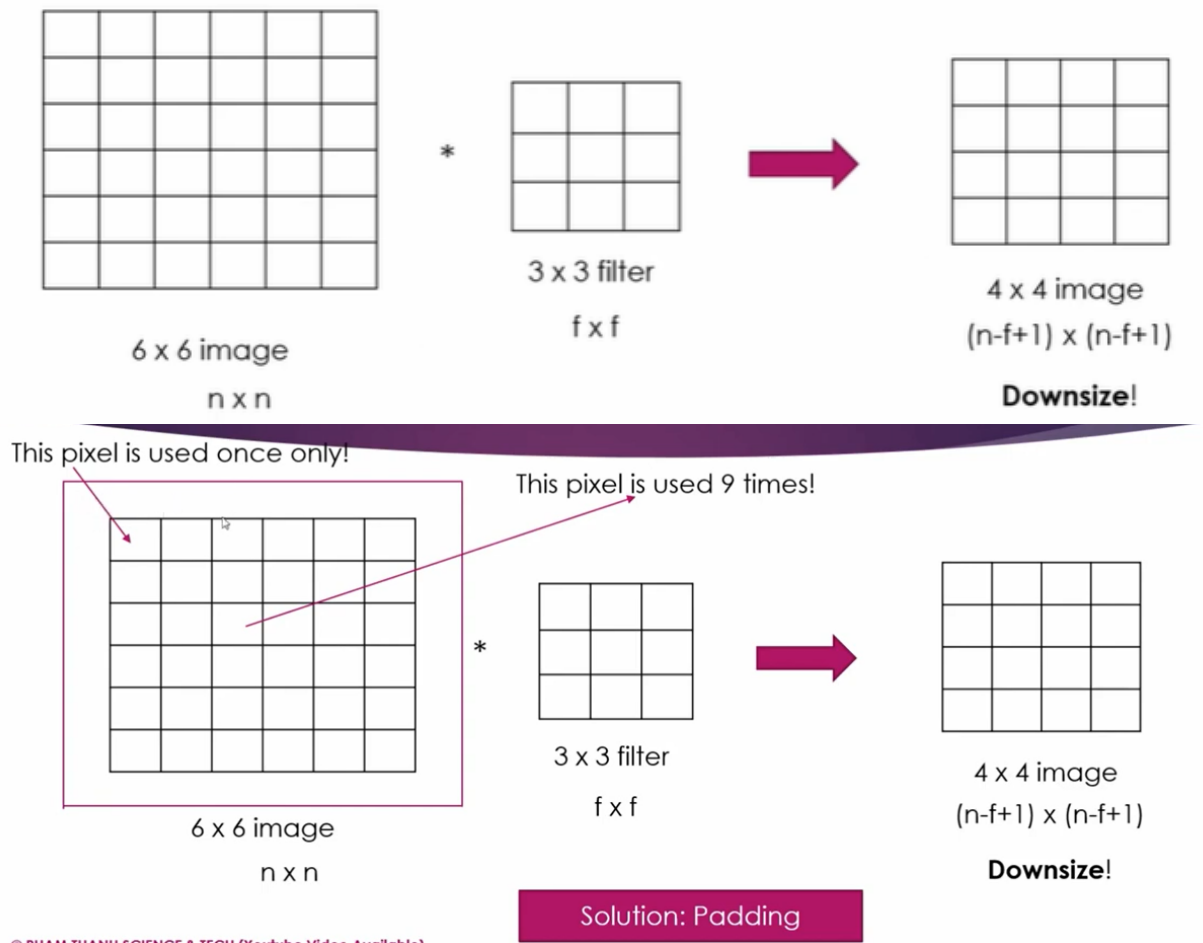
Ban đầu chúng ta đưa vào 1 bức ảnh có tỉ lệ 6x6 thực hiện phép tích chập với bộ lọc 3x3 trong đó các giá trị của từng ô được đánh trọng số từ W1-9 , thì khi mới huấn luyện 9 bộ số này được thiết lập một cách ngẫu nhiên và qua nhiều bước huấn luyện trên 1 mạng nơ ron thì đến gần cuối của kết quả đầu ra các giá trị sẽ được thay đổi làm sao cho được 1 kết quả tối ưu để làm sao nhận dạng các góc cạnh 1 cách chính xác nhất để từ đây xác định được các chi tiết của các vật thể



Với mỗi bài toán thì trong 1 mạng nơ ron có thể có rất nhiều các filter có tập giá trị khác nhau để phân biệt được các góc cạnh theo bài toán cụ thể.

2.8.5.2.Padding.

Padding : là một kỹ thuật được sử dụng để thêm các giá trị không gian (zero-padding) vào xung quanh đầu vào của một lớp tích chập.



Hình 2.8.5.2.1: tổng quan các ma trận.

Có thể thấy việc chúng ta xét từng vị trí như vậy thì những pixel ở hàng đầu ảnh nó đóng góp rất ít thông tin cho bức ảnh output khiến cho bức ảnh đầu ra của chúng ta có thể mất mát thông tin đặc trưng quan trọng ở gần biên ảnh đầu vào, và để khắc phục điều đó chúng ta sẽ dùng kỹ thuật padding.

Chúng ta sẽ thêm các giá trị 0 hoặc các giá trị biên vào xung quanh ảnh hoặc đầu vào để tăng kích thước của chúng trước khi áp dụng phép tích chập.

Tuy nhiên chúng ta không thể lạm dụng padding nhiều nó sẽ dẫn đến vấn đề nhiễu, khi chúng ta padding chúng ta đưa các giá trị 0 vào biên xung quanh mà những giá trị đây ko phải của bức ảnh gốc đầu vào và nó không phải là thông tin của bức ảnh gốc và nó thực sự có thể làm nhiễu thông tin thực sự của bức ảnh gốc này. Được cái này và sẽ mất cái kia : chúng ta muốn giữ được kích thước hay đóng góp thêm thông tin của pixel có đặc trưng ở biên thì sẽ xảy ra vấn đề bị nhiễu.

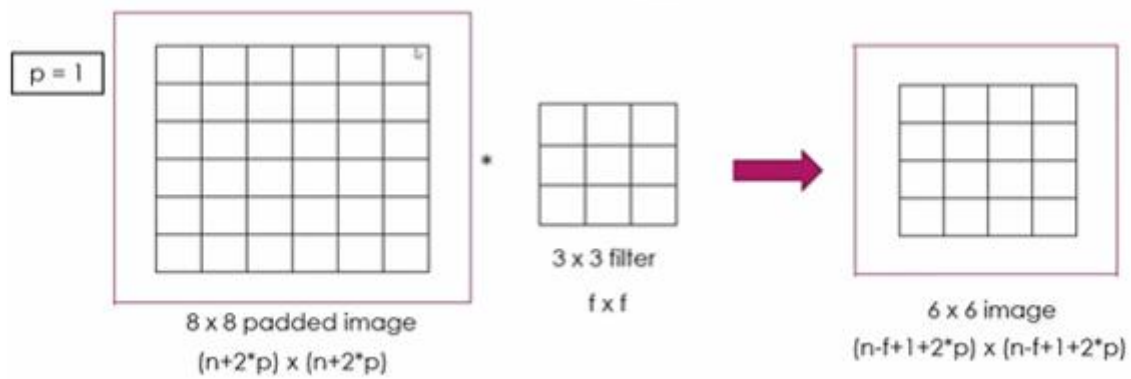
Padding không nhất thiết $p=1$, có thể $= 2, 3, \dots$

Same và valid tích chập:

Valid : No padding : Output : $(n-f+1) \times (n-f+1)$

Same: Output size = input Size: Output = $(n-f+1+2p) \times (n-f+1+2p)$

Cách đánh padding phù hợp với yêu cầu output là Same: $n-f+1+2p = n \Rightarrow 2p = f - 1$

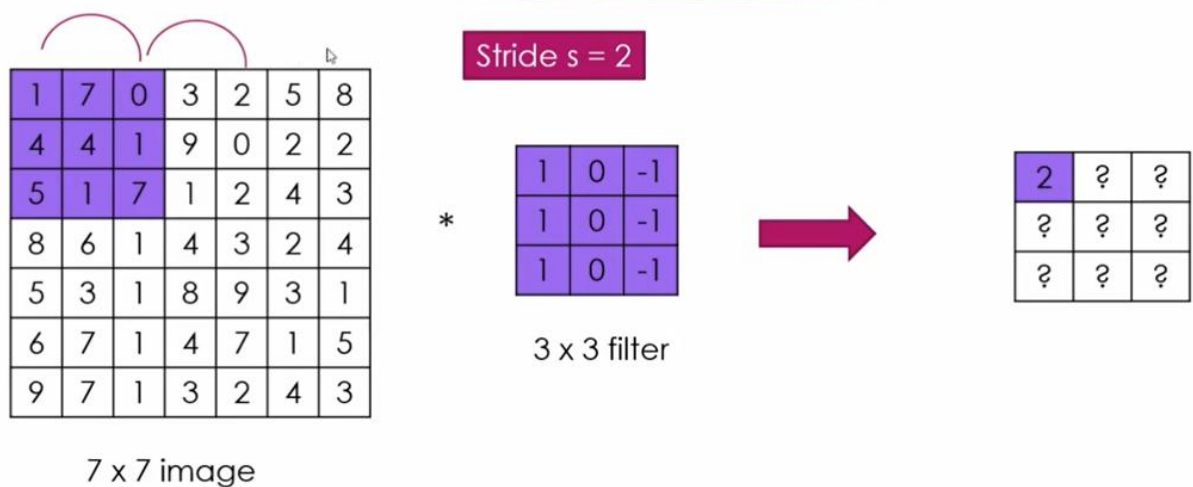


Hình 2.8.5.2.2: Cách đánh padding.

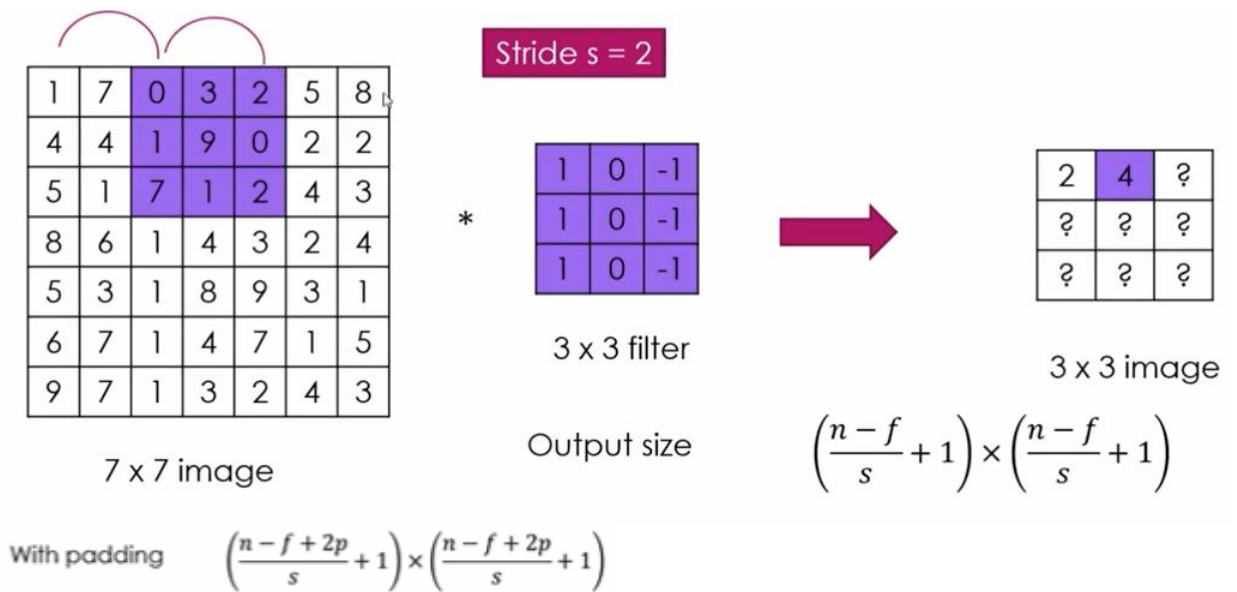
2.8.5.2.Stride (bước nhảy).

Stride (bước nhảy): là một tham số quan trọng được sử dụng để xác định cách mà bộ lọc di chuyển trên Input và áp dụng phép tích chập. có vai trò chính:

- Giảm kích thước đầu ra (Output) và giảm số lượng tham số cần học trong mạng nơ ron
- Giảm tính toán: giảm số lượng phép tính toán cần thiết và giúp tăng tốc độ huấn luyện của suy luận của mạng



Hình 2.8.5.2.1: bước chập 1.

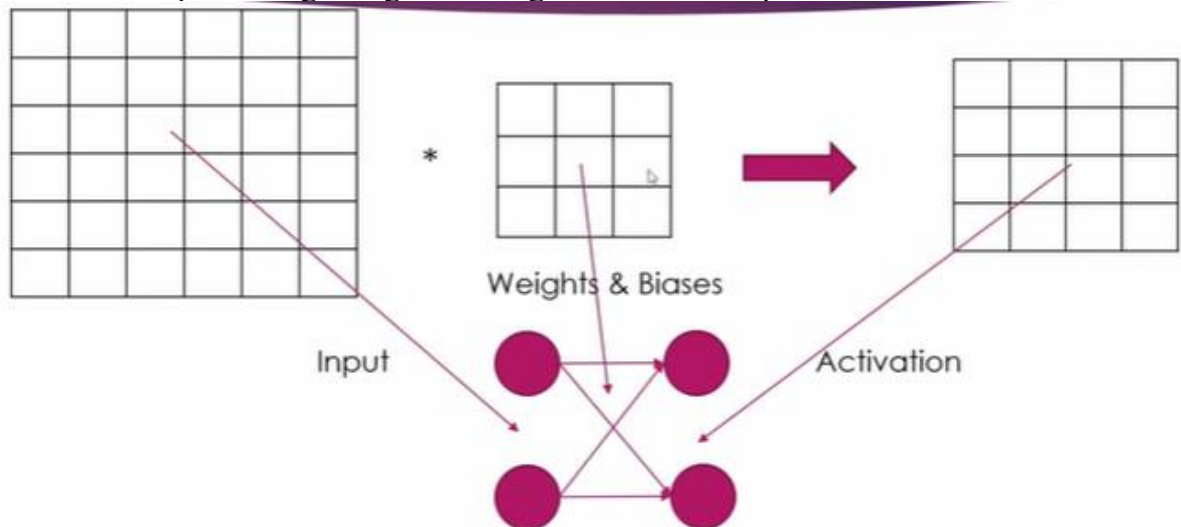


Hình 2.8.5.2.2: bước chập 2.

Nhận xét: trường hợp $(n-f)/s$ là 1 số lẻ phải thì chúng ta sẽ làm tròn đến hàng đơn vị (ví dụ: 1,5 \Rightarrow 1)

Tổng kết về phép tích chập: $\left\lfloor \frac{n-f+2p}{s} + 1 \right\rfloor \times \left\lfloor \frac{n-f+2p}{s} + 1 \right\rfloor$

Mô hình khái quát tương trưng cho mạng nơ ron tích chập:

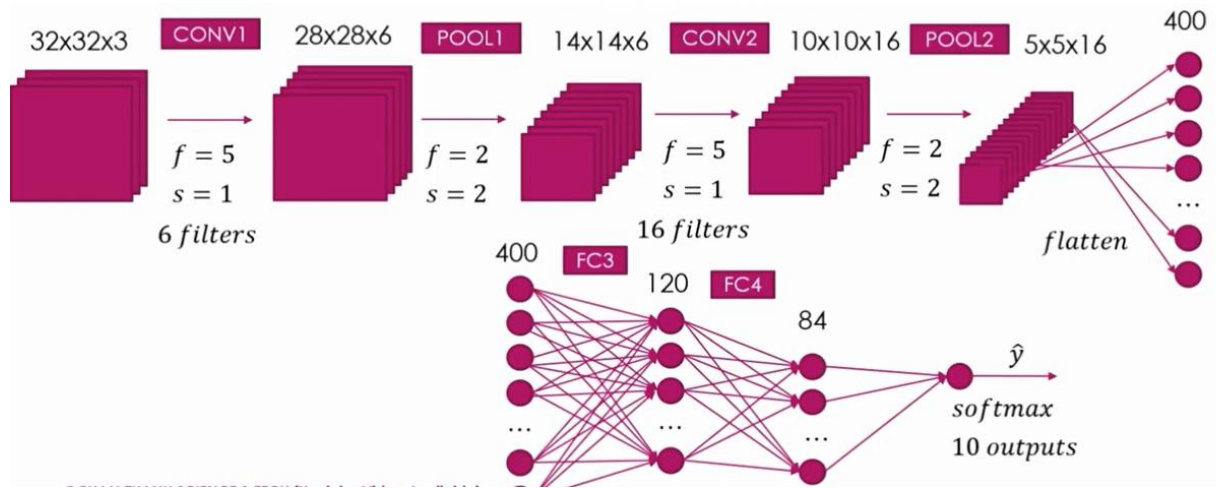


Hình 2.8.5.2.3: minh họa 1 nút CNN.

2.2.6. Mạng LeNet-5 và bài toán nhận diện ký tự số.

Mạng LeNet-5 là một kiến trúc mạng nơ-ron tích chập (CNN) đầu tiên được giới thiệu bởi Yann LeCun, Léon Bottou, Yoshua Bengio và Patrick Haffner vào năm 1998. Mạng LeNet-5 được thiết kế đặc biệt để nhận dạng ký tự viết tay trong bộ dữ liệu nhận dạng ZIP code của Hoa Kỳ. Nó đã trở thành một cơ sở cho các kiến trúc CNN tiếp theo và góp phần vào thành công của deep learning trong lĩnh vực thị giác máy tính.

Chúng ta sẽ áp dụng bài toán lên mạng LeNet này:



Hình 2.8.6.1: các bước cnn tích chập.

Tóm tắt công thức tính kích thước ảnh Output: $\left\lfloor \frac{n-f+2p}{s} + 1 \right\rfloor \times \left\lfloor \frac{n-f+2p}{s} + 1 \right\rfloor$

Đầu tiên, Input và 1 bức ảnh RGB (3 kênh), lớp tích chập đầu tiên (CONV1) với kích thước filter =5 (f=5), stride = 1 (s=1) và có 6 filters (6 kênh filter)

Ta có: $n=32 \Rightarrow \left\lfloor \frac{32-5+2 \times 0}{1} + 1 \right\rfloor = 28$. Vậy kích thước cho ảnh Output là 28x28x6 (số channel = số filter = 6)

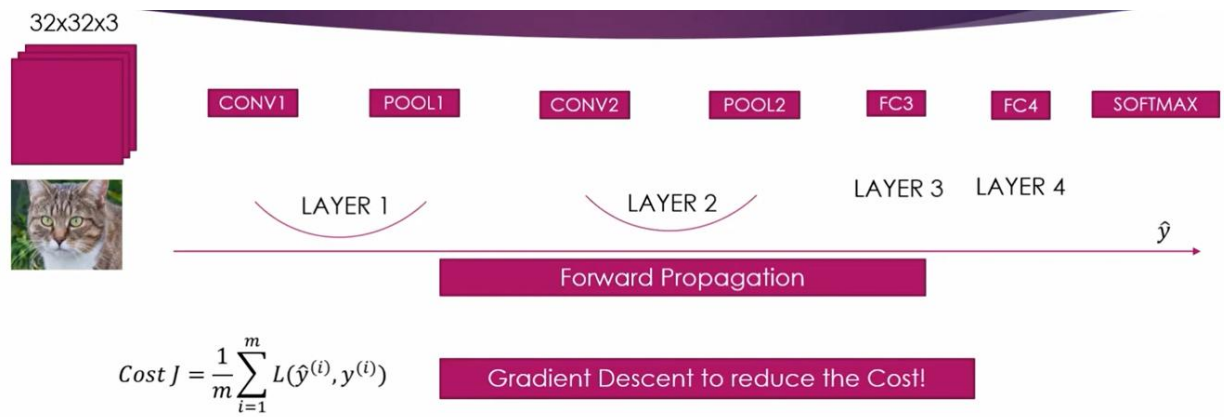
Tiếp theo, chúng ta sẽ đến layer Pooling , pooling sẽ giảm kích thước ảnh xuống nhưng sẽ giữ được số kênh của ảnh mà chúng ta đã huấn luyện qua layer tích chập trước đó.

	Activation Shape	Activation Size	# parameters
Input	(32, 32, 3)	3,072	0
CONV1 (f=5, s=1)	(28, 28, 8)	6,272	208
POOL1	(14, 14, 8)	1,568	0
CONV2 (f=5, s=1)	(10, 10, 16)	1,600	416
POOL2	(5, 5, 16)	400	0
FC3	(120, 1)	120	48,001
FC4	(84, 1)	84	10,081
SOFTMAX	(10, 1)	10	841
TOTAL			59,547

Hình 2.8.6.2:thông số của mạng cnn.

Kết luận:

- Chúng ta có được mô hình tổng quát khái quát quá trình hoạt động của 1 mạng nơ ron tích chập những bước chúng ta vừa làm ở trên chỉ là bước đầu cho việc đưa một bức ảnh Input và có được 1 bức ảnh Output xong chúng ta còn phải kiểm tra rằng bức ảnh Output này liệu có gần với Output mong muốn của chúng ta hay không. Sau đó, chúng ta tính hàm Cost như một mạng nơ ron thông thường, sau tính hàm Cost xong chúng ta thực hiện Gradient Descent để từ đây chúng ta hiệu chỉnh lại các tham số và giảm Cost xuống khi Cost giảm xuống mức chấp nhận được thì chúng ta sẽ kết thúc quá trình training thì chúng ta sử dụng để dự đoán.



Hình 2.8.6.3: các bước layer.

2.3. Mạng phần dư (ResNet).

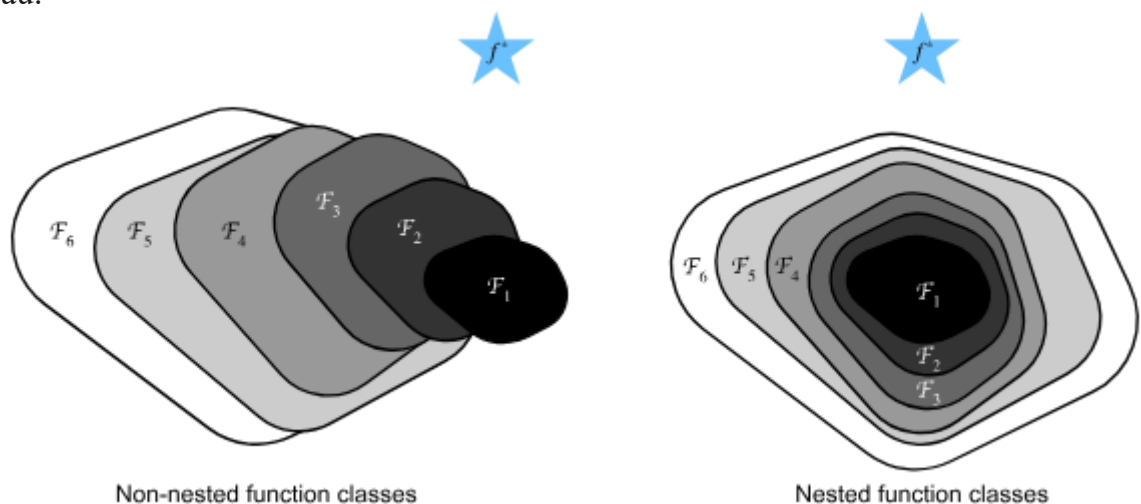
2.3.1. Tổng quan về ResNet.

ResNet là một mạng tích chập mạnh mẽ được thiết kế để làm việc với hàng trăm lớp, một kiến trúc mạng nơ-ron học sâu được thiết kế để giải quyết vấn đề vanishing gradient trong các mạng nơ-ron sâu, thông qua các kết nối nhảy (skip connections).

- Thông thường đơn giản chúng ta chỉ nghĩ nếu tăng số lượng layer trong các bước tích chập lên thì độ chính xác của mạng tăng lên, nhưng thực tế nó lại ngược lại và dẫn đến hiện tượng lỗi đào tạo cao hơn xảy ra. Chúng ta xét ví dụ: Gọi F là lớp các hàm số mà kiến trúc mạng N có thể đạt được, với mọi $f \in F$ luôn tồn tại một bộ tham số W cho kiến trúc mạng N được huấn luyện trên bộ dữ liệu phù hợp. Giả sử hàm số f^* là hàm cần tìm.

$$f_F^* \stackrel{\text{def}}{=} \operatorname{argmin}_{f \in F} L(\mathbf{X}, \mathbf{y}, f) \text{ subject to } f \in F$$

Nếu chúng ta thiết kế F' mạnh hơn thì kỳ vọng rằng độ chính xác của mô hình sẽ tăng lên hay tham số $f_{F'}^*$ sẽ tốt hơn tham số f_F^* . Tuy nhiên, điều này không thể khẳng định được nếu như $F \not\subseteq F'$. Trên thực tế, đôi khi chính hàm f_F^* , thậm chí còn tệ hơn. Chúng ta minh họa như sau:



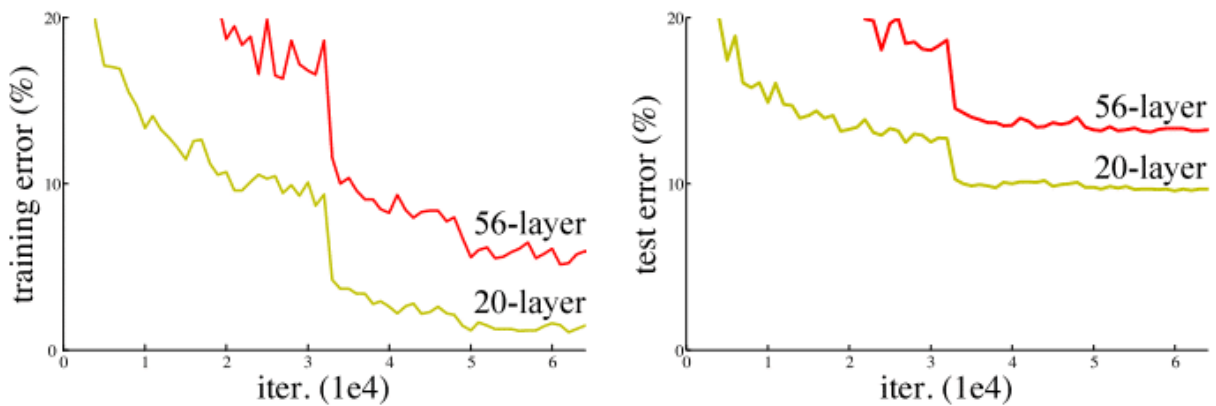
Bên trái minh họa cho các lớp hàm số không lồng nhau cho nên nếu chúng ta tăng kích thước của mạng thậm chí làm cho các hàm f^* có khoảng cách ngày càng xa hơn khi mở rộng và điều này sẽ không xảy ra đối với các hàm số lồng nhau như hình bên trái. Chỉ khi

các lớp hàm lớn hơn chứa các lớp nhỏ hơn, thì mới đảm bảo rằng việc tăng thêm các tầng sẽ tăng khả năng biểu diễn của mạng.

Ý tưởng trọng tâm của ResNet là mỗi tầng được thêm vào nên có một thành phần là hàm số đồng nhất. Điều này có nghĩa rằng, nếu ta huấn luyện tầng mới được thêm vào thành một ánh xạ đồng nhất $f(x) = x$ thì mô hình mới sẽ hiệu quả ít nhất bằng mô hình ban đầu. Vì tầng được thêm vào có thể khớp dữ liệu huấn luyện tốt hơn, dẫn đến sai số huấn luyện cũng nhỏ hơn.

2.3.2. Vanishing gradient.

Vanishing gradient Là vấn đề xảy ra khi huấn luyện các mạng nơ ron nhiều lớp. Khi huấn luyện, giá trị đạo hàm là thông tin phản hồi của quá trình lan truyền ngược. Giá trị này trở nên vô cùng nhỏ tại các lớp nơ ron đầu tiên khiến cho việc cập nhật trọng số mạng không thể xảy ra. Vì độ dốc được truyền ngược trở lại các lớp trước đó, phép nhân lặp đi lặp lại có thể làm cho độ dốc cực nhỏ. Kết quả là, hiệu suất của mạng bị bão hòa hoặc giảm hiệu quả nhanh chóng.



Thông thường chúng ta sẽ có một hyperparameter (số epoch là số lần mà training set được duyệt qua một lần và weights được cập nhật) định nghĩa cho số lượng vòng lặp để thực hiện quá trình này. Nếu số lượng vòng lặp quá nhỏ thì ta gặp phải trường hợp mạng có thể sẽ không cho ra kết quả tốt và ngược lại thời gian training sẽ lâu nếu số lượng vòng lặp quá lớn. Trong thực tế Gradients thường sẽ có giá trị nhỏ dần khi đi xuống các layer thấp hơn. Dẫn đến kết quả là các cập nhật thực hiện bởi Gradients Descent không làm thay đổi nhiều weights của các layer đó và làm chúng học nhưng không đem lại kết quả tốt và không học được hết các đặc trưng.

2.3.3. Skip connection.

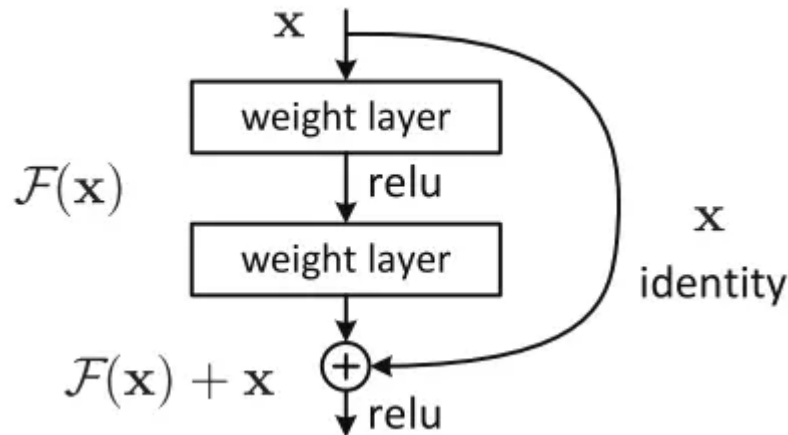
Ý tưởng chính của ResNet giúp mạng có thể đào tạo sâu hơn mà không khiến hiện tượng vanishing gradient xảy ra trong quá trình Backpropagation. Giả sử output của layer nông là x . Trong quá trình forward của mạng nó được đưa qua một phép biến đổi tuyến tính $F(x)$. Chúng ta giả sử output của phép biến đổi tuyến tính này là $H(x)$. Một residual (phần dư) giữa deep layer và shallow layer là

$$F(x; W_i) := H(x) - x$$

Trong đó W_i là các tham số của mô hình CNN với phép biến đổi F và nó được tối ưu trong quá trình huấn luyện.

2.3.4. Kiến trúc mạng ResNet.

Chúng ta biết được khi CNN phát triển sâu hơn, độ dốc biến mất có xu hướng xảy ra, điều này tác động tiêu cực đến hiệu suất mạng, sự cố biến mất độ dốc xảy ra khi độ dốc được truyền ngược trở lại các lớp trước đó dẫn đến độ dốc rất nhỏ, cho nên ResNet sử dụng skip connection để có thể giải quyết được vấn đề Vanishing Gradient. Và một khối dư được thể hiện như hình sau:

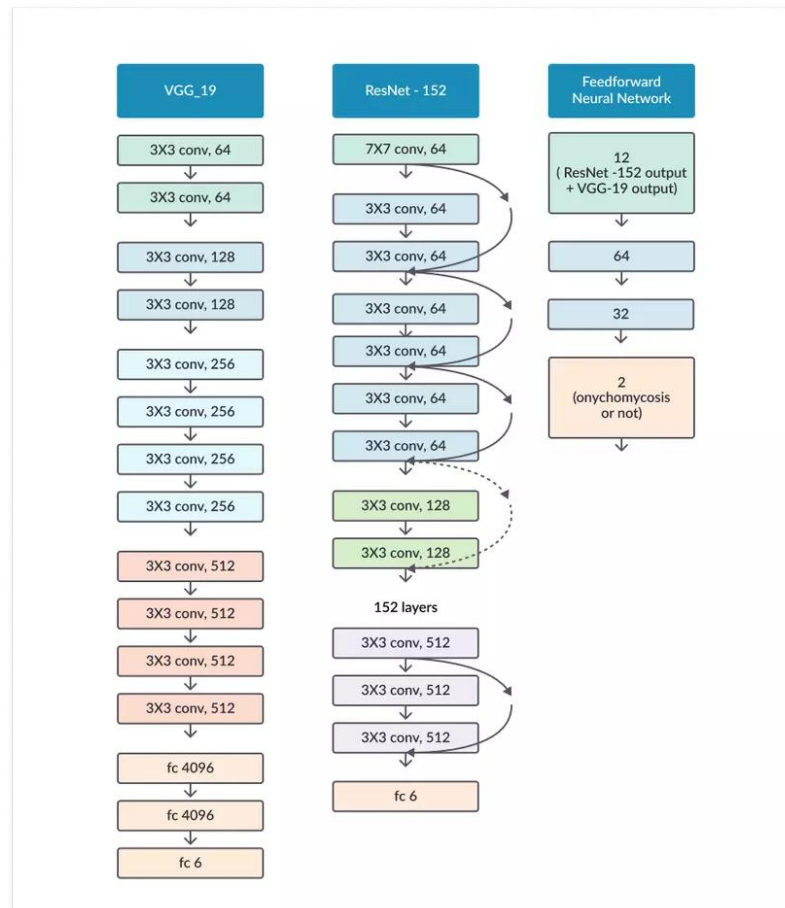


ResNet gần như tương tự với các mạng gồm có convolution, pooling, activation và fully-connected layer. Ảnh bên trên hiển thị khối dư được sử dụng trong mạng. Xuất hiện một mũi tên cong xuất phát từ đầu và kết thúc tại cuối khối dư. Hay nói cách khác là sẽ bổ sung Input X vào đầu ra của layer, hay chính là phép cộng mà ta thấy trong hình minh họa, việc này sẽ chống lại việc đạo hàm bằng 0, do vẫn còn cộng thêm X . Với $H(x)$ là giá trị dự đoán, $F(x)$ là giá trị thật (nhãn), chúng ta muốn $H(x)$ bằng hoặc xấp xỉ $F(x)$. Ta có:

$$F(x) = X \rightarrow \text{weight1} \rightarrow \text{ReLU} \rightarrow \text{weight2}$$

$$H(x) = F(x) + x \rightarrow \text{ReLU}$$

Như chúng ta đã biết việc tăng số lượng các lớp trong mạng làm giảm độ chính xác, nhưng muốn có một kiến trúc mạng sâu hơn có thể hoạt động tốt.

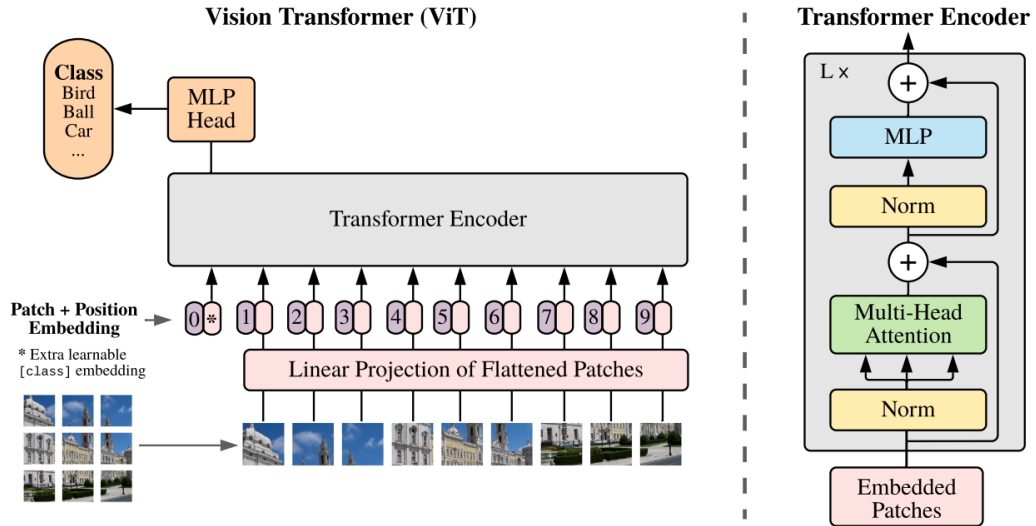


Chúng ta có thể nhận xét được so với mô hình VGG_19 thì mô hình ResNet_152 có thể học được nhiều hơn giúp gradient truyền qua mạng mà không bị suy giảm, giúp quá trình huấn luyện ổn định hơn. So với VGG thì chỉ 2 lớp 64 còn đối với ResNet áp dụng skip connection có thể học lên tới 7 lớp, để chúng ta thấy được sức mạnh của mạng ResNet so với những kiến trúc mạng trước khi ResNet ra đời.

2.4. Vision Transformer (ViT).

2.4.1. Kiến trúc Transformers.

- Văn bản được chuyển đổi thành các biểu diễn số được gọi là mã hóa bởi mô hình hình ngôn ngữ lớn (LLM) và mỗi mã được chuyển đổi embedding biểu diễn dưới dạng ma trận vecto.



Hình 1: Tổng quan mô hình. chia một hình ảnh thành các mảnh có kích thước cố định, nhúng tuyến tính cho từng mảnh, thêm vị trí nhúng, và đưa chuỗi vector thu được vào bộ mã hóa Transformer tiêu chuẩn. Để thực hiện phân loại, sử dụng phương pháp tiêu chuẩn là thêm một "token phân loại" có thể học được vào chuỗi. Minh họa bộ mã hóa Transformer được lấy cảm hứng từ Vaswani và cộng sự (2017).

- Tổng quan về mô hình được minh họa trong Hình 1. Transformer tiêu chuẩn nhận đầu vào là một chuỗi một chiều các token nhúng. Để xử lý ảnh 2D, định hình lại hình ảnh $x \in R^{H \times W \times C}$ thành một chuỗi các mảnh 2D được trải phẳng $x_p \in R^{N \times (P^2 \cdot C)}$, trong đó (H, W) là độ phân giải của hình ảnh gốc, C là số kênh, (P, P) là độ phân giải của từng mảnh ảnh, và $N = HW/P^2$ là số lượng mảnh thu được, đồng thời cũng là chiều dài chuỗi đầu vào hiệu quả cho Transformer. Transformer sử dụng kích thước vector tiềm ẩn không đổi D qua tất cả các lớp, vì vậy trải phẳng các mảnh và ánh xạ sang chiều D bằng một phép chiếu tuyến tính có thể học được (Phương trình 1). Gọi đầu ra của phép chiếu này là các patch nhúng.
- Tương tự như token [class] của BERT, thêm một embedding có thể học được vào đầu chuỗi các patch nhúng ($z_0^0 = x_{class}$), trạng thái của nó tại đầu ra của bộ mã hóa Transformer (z_L^0) phục vụ như biểu diễn ảnh y (Phương trình 4). Cả trong quá trình tiền huấn luyện và tinh chỉnh, một đầu phân loại được gắn vào z_L^0 . Đầu phân loại được thực hiện bằng một MLP với một lớp ẩn trong quá trình tiền huấn luyện và bằng một lớp tuyến tính duy nhất trong quá trình tinh chỉnh.
- Các vị trí nhúng được thêm vào các patch nhúng để giữ lại thông tin vị trí. Sử dụng các vị trí nhúng 1D có thể học được tiêu chuẩn, vì chưa nhận thấy các cải thiện hiệu năng đáng kể từ việc sử dụng các vị trí nhúng 2D tiên tiến hơn (Phụ lục D.4). Chuỗi các vector nhúng thu được sẽ được sử dụng làm đầu vào cho bộ mã hóa.
- Bộ mã hóa Transformer (Vaswani và cộng sự, 2017) bao gồm các lớp xen kẽ của cơ chế tự chú ý đa đầu (MSA, xem Phụ lục A) và các khối MLP (Phương trình 2, 3). Chuẩn hóa lớp (LN) được áp dụng trước mỗi khối, và các kết nối dư được thêm sau mỗi khối (Wang và cộng sự, 2019; Baevski & Auli, 2019).

$$z_0 = [x_{class}; x_p^1 E; x_p^2 E; \dots; x_p^N E] + E_{pos} \quad E \in R^{(P^2.C) \times D}, E_{pos} \in R^{(N+1) \times D} \quad (1)$$

$$z'_l = MSA(LN(z_{l-1})) + z_{l-1} \quad l = 1 \dots L \quad (2)$$

$$z_l = MSA(LN(z'_l)) + z'_l \quad l = 1 \dots L \quad (3)$$

$$y = LN(z_L^o) \quad (4)$$

-**Thiên vị quy nạp.** lưu ý rằng Vision Transformer có ít thiên vị đặc thù về hình ảnh hơn nhiều so với CNN. Trong CNN, tính địa phương, cấu trúc láng giềng hai chiều và tính đẳng biến dịch được tích hợp sẵn trong từng lớp trên toàn bộ mô hình. Trong ViT, chỉ các lớp MLP là địa phương và đẳng biến dịch, trong khi các lớp tự chú ý là toàn cục. Cấu trúc láng giềng hai chiều được sử dụng rất hạn chế: ở đầu mô hình bằng cách cắt ảnh thành các mảnh và trong quá trình tinh chỉnh để điều chỉnh các vị trí nhúng cho các hình ảnh có độ phân giải khác nhau (như mô tả dưới đây). Ngoài ra, các vị trí nhúng tại thời điểm khởi tạo không mang theo bất kỳ thông tin nào về vị trí 2D của các mảnh và tất cả các mối quan hệ không gian giữa các mảnh phải được học hoàn toàn từ đầu.

-**Kiến trúc Lai.** Như một giải pháp thay thế cho các mảnh ảnh thô, chuỗi đầu vào có thể được hình thành từ các bản đồ đặc trưng của CNN (LeCun và cộng sự, 1989). Trong mô hình lai này, phép chiếu nhúng mảnh E (Phương trình 1) được áp dụng cho các mảnh được trích xuất từ bản đồ đặc trưng CNN. Như một trường hợp đặc biệt, các mảnh có thể có kích thước không gian 1x1, có nghĩa là chuỗi đầu vào được thu được bằng cách trải phẳng các chiều không gian của bản đồ đặc trưng và chiếu sang chiều Transformer. Embedding đầu vào phân loại và vị trí nhúng được thêm vào như mô tả ở trên.

2.5. Zero-Shot Learning.

- Zero-Shot Learning là một kỹ thuật cho phép các mô hình được đào tạo trước dự đoán nhãn lớp của dữ liệu chưa biết trước đó, tức là các mẫu dữ liệu không có trong dữ liệu đào tạo. Ví dụ, một mô hình học sâu (DL) được đào tạo để phân loại sư tử và hổ có thể phân loại chính xác một con hổ bằng cách sử dụng học không-shot mặc dù không tiếp xúc với hổ trong quá trình đào tạo. Điều này đạt được bằng cách tận dụng các mối quan hệ ngữ nghĩa hoặc thuộc tính (như môi trường sống, loại da, màu sắc, v.v.) liên quan đến các lớp, thu hẹp khoảng cách giữa các danh mục đã biết và chưa biết.
- Học tập không-shot đặc biệt có giá trị trong các lĩnh vực như thị giác máy tính (CV) và xử lý ngôn ngữ tự nhiên (NLP), nơi quyền truy cập vào các tập dữ liệu được gắn nhãn bị hạn chế. Các nhóm có thể chú thích các tập dữ liệu lớn bằng cách tận dụng các mô hình học tập không-shot, đòi hỏi nỗ lực tối thiểu từ các chuyên gia chuyên ngành để gắn nhãn dữ liệu cụ thể cho từng lĩnh vực. Ví dụ, ZSL có thể giúp tự động hóa chú thích hình ảnh y tế để chẩn đoán hiệu quả hoặc tìm hiểu các mẫu DNA phức tạp từ dữ liệu y tế chưa được gắn nhãn.

- Điều quan trọng là phải phân biệt zero-shot learning với one-shot learning và few-shot learning. Trong one-shot learning, một mẫu có sẵn cho mỗi lớp chưa thấy. Trong few-shot learning, một số lượng nhỏ mẫu có sẵn cho mỗi lớp chưa thấy. Mô hình tìm hiểu thông tin về các lớp này từ dữ liệu hạn chế này và sử dụng nó để dự đoán nhãn cho các mẫu chưa thấy.

2.5.1. Các loại học tập Zero-Shot:

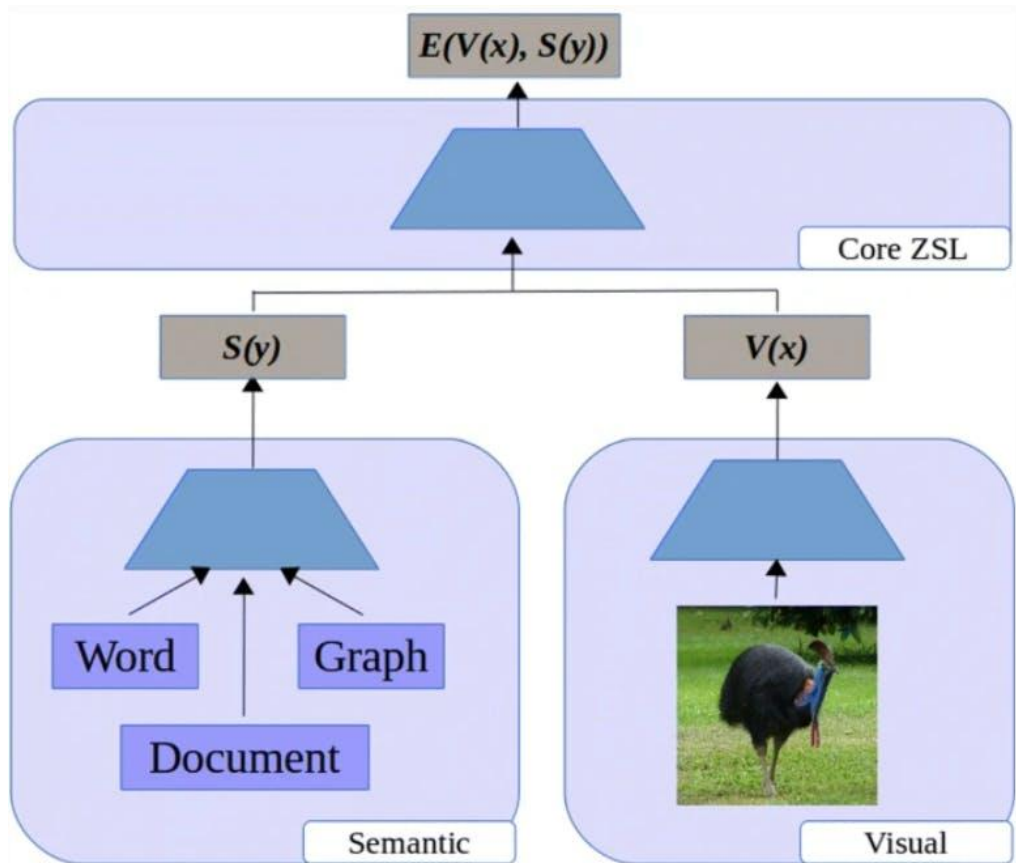
- Học Zero-Shot dựa trên thuộc tính:
 - o ZSL dựa trên thuộc tính liên quan đến việc đào tạo một mô hình phân loại bằng cách sử dụng các thuộc tính cụ thể của dữ liệu được gắn nhãn. Các thuộc tính đề cập đến các đặc điểm khác nhau trong dữ liệu được gắn nhãn, chẳng hạn như màu sắc, hình dạng, kích thước, v.v. Một mô hình ZSL có thể suy ra nhãn của các lớp mới bằng cách sử dụng các thuộc tính này nếu lớp mới đủ giống với các lớp thuộc tính trong dữ liệu đào tạo.
- Học tập Zero-Shot dựa trên nhúng ngữ nghĩa:
 - o Nhúng ngữ nghĩa là các biểu diễn vector của các thuộc tính trong không gian ngữ nghĩa, tức là thông tin liên quan đến ý nghĩa của từ, n-gram và cụm từ trong văn bản hoặc hình dạng, màu sắc và kích thước trong hình ảnh. Ví dụ, nhúng hình ảnh hoặc từ là một vector có chiều cao, trong đó mỗi phần tử biểu diễn một thuộc tính cụ thể. Các phương pháp như Word2Vec, GloVe và BERT thường được sử dụng để tạo nhúng ngữ nghĩa cho dữ liệu văn bản. Các mô hình này tạo ra các vector có chiều cao, trong đó mỗi phần tử có thể biểu diễn một thuộc tính ngôn ngữ hoặc ngữ cảnh cụ thể.
 - o Các mô hình học Zero-shot có thể học các nhúng ngữ nghĩa này từ dữ liệu được gắn nhãn và liên kết chúng với các lớp cụ thể trong quá trình đào tạo. Sau khi được đào tạo, các mô hình này có thể chiếu các lớp đã biết và chưa biết lên không gian nhúng này. Bằng cách đo độ tương đồng giữa các nhúng bằng các phép đo khoảng cách, mô hình có thể suy ra loại dữ liệu chưa biết.
 - o Một số phương pháp ZSL dựa trên nhúng ngữ nghĩa đáng chú ý là Semantic AutoEncoder (SAE), DeViSE và VGSE.
 - o SAE liên quan đến một khuôn khổ mã hóa-giải mã phân loại các đối tượng chưa biết bằng cách tối ưu hóa hàm tái tạo bị hạn chế.
 - o Tương tự như vậy, DeViSE đào tạo một mô hình nhúng ngữ nghĩa trực quan sâu để phân loại hình ảnh chưa biết thông qua thông tin ngữ nghĩa dựa trên văn bản.
 - o VGSE tự động học nhúng ngữ nghĩa của các mảng hình ảnh, yêu cầu chú thích tối thiểu ở cấp độ con người và sử dụng mô-đun quan hệ lớp để tính toán điểm tương đồng giữa nhúng lớp đã biết và chưa biết để học không cần thực hiện cú đánh nào.
- Học tập Zero-Shot tổng quát (GZSL):
 - o GZSL mở rộng kỹ thuật học zero-shot truyền thống để mô phỏng khả năng nhận dạng của con người. Không giống như ZSL truyền thống, chỉ tập trung vào các lớp chưa biết, GZSL đào tạo các mô hình trên các lớp đã biết và chưa biết trong quá trình học có giám sát. Bạn đào tạo các mô hình GSZL bằng cách thiết lập mối quan hệ giữa các lớp đã biết và chưa biết, tức là chuyển

kiến thức từ các lớp đã biết sang các lớp chưa biết bằng cách sử dụng các thuộc tính ngữ nghĩa của chúng. Một kỹ thuật bổ sung cho cách tiếp cận này là thích ứng miền.

- Thích ứng miền là một kỹ thuật học chuyển giao hữu ích về mặt này. Nó cho phép các học viên AI tái sử dụng một mô hình được đào tạo trước cho một tập dữ liệu khác chứa dữ liệu chưa được gắn nhãn bằng cách chuyển thông tin ngữ nghĩa.
- Các nhà nghiên cứu Pourpanah, Farhad và cộng sự đã trình bày một đánh giá toàn diện về các phương pháp GZSL. Họ phân loại GZSL thành hai loại dựa trên cách kiến thức được chuyển giao và học từ các lớp đã biết sang các lớp chưa biết:
 - Các phương pháp dựa trên nhúng: Thường dựa trên cơ chế chú ý, bộ mã hóa tự động, đồ thị hoặc học hai chiều. Các phương pháp như vậy học các biểu diễn ngữ nghĩa cấp thấp hơn bắt nguồn từ các đặc điểm trực quan của các lớp đã biết trong tập huấn luyện và phân loại các mẫu chưa biết bằng cách đo độ tương đồng của chúng với các biểu diễn của các lớp đã biết.
 - Các phương pháp dựa trên sinh sản: Các kỹ thuật này thường bao gồm Mạng đối nghịch sinh sản (GAN) và Bộ mã hóa tự động biến thể (VAE). Chúng học các biểu diễn trực quan từ các tính năng lớp đã biết và nhúng từ từ các mô tả lớp đã biết và chưa biết để đào tạo một mô hình sinh sản có điều kiện nhằm tạo các mẫu đào tạo. Quy trình này đảm bảo tập đào tạo bao gồm các lớp đã biết và chưa biết, biến việc học không cân bằng thành một vấn đề học có giám sát.
- Học tập Zero-Shot đa phương thức:
 - ZSL đa phương thức kết hợp thông tin từ nhiều phương thức dữ liệu, chẳng hạn như văn bản, hình ảnh, video và âm thanh, để dự đoán các lớp chưa biết. Ví dụ, bằng cách đào tạo một mô hình sử dụng hình ảnh và mô tả văn bản liên quan của chúng, một học viên ML có thể trích xuất các nhúng ngữ nghĩa và phân biệt các liên kết có giá trị. Mô hình có thể trích xuất các nhúng ngữ nghĩa và tìm hiểu các liên kết có giá trị từ dữ liệu này. Với khả năng zero-shot, mô hình này có thể khái quát hóa thành các tập dữ liệu chưa biết tương tự với hiệu suất dự đoán chính xác.

2.5.2. Kiến trúc cơ bản của Zero-Shot Learning.

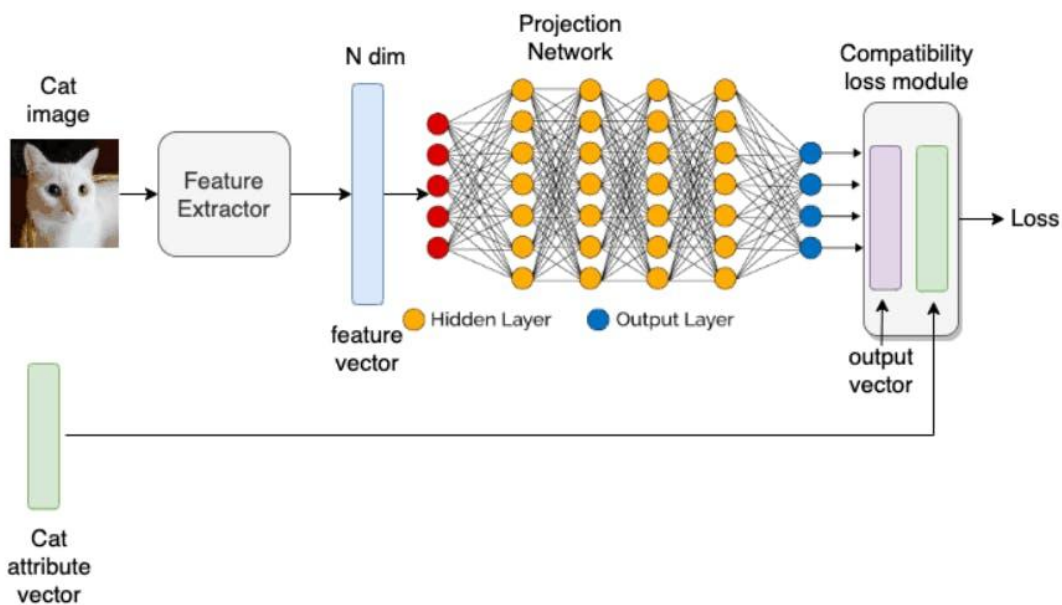
- Hãy xem xét một mô hình phân loại hình ảnh ZSL. Về cơ bản, nó bao gồm các mô-đun nhúng ngữ nghĩa và trực quan và một thành phần học zero-shot tính toán sự giống nhau giữa hai nhúng.



- Giải thích:
 - Core ZSL: Trung tâm của mô hình, nơi hai nguồn thông tin (ngữ nghĩa và hình ảnh) được kết hợp.
 - Ngữ nghĩa ($S(y)$):
 - Bao gồm Word, Graph, và Document, chỉ ra rằng mô hình sử dụng các loại thông tin khác nhau để hiểu ngữ nghĩa của các lớp đối tượng.
 - Word: Từ khóa hoặc mô tả ngắn.
 - Graph: Các mối quan hệ giữa các đối tượng hay thuộc tính.
 - Document: Văn bản chi tiết liên quan đến đối tượng.
 - Hình ảnh ($V(x)$):
 - Đại diện cho dữ liệu hình ảnh mà mô hình sẽ xử lý, như hình ảnh của một con đà điểu trong ví dụ.
 - $E(V(x), S(y))$: Đây có thể là một hàm ánh xạ, giúp mô hình kết nối các thông tin hình ảnh và ngữ nghĩa để phân loại và nhận diện đối tượng mà không cần dữ liệu đã gắn nhãn cho lớp đó.
- Mô-đun nhúng ngữ nghĩa chiếu thông tin dạng văn bản hoặc dựa trên thuộc tính, như tài liệu, biểu đồ kiến thức hoặc mô tả hình ảnh, lên không gian vector nhiều chiều.
- Tương tự như vậy, mô-đun nhúng trực quan chuyển đổi dữ liệu trực quan thành các nhúng nắm bắt các thuộc tính cốt lõi của hình ảnh. Cả nhúng ngữ nghĩa và nhúng trực quan đều được chuyển đến mô-đun ZSL để tính toán mức độ tương đồng của chúng và tìm hiểu mối quan hệ giữa chúng.

2.5.3. Quá trình học tập Zero-shot.

- Quá trình học tập bao gồm việc giảm thiểu hàm mất mát được điều chỉnh theo trọng số của mô hình trên các ví dụ đào tạo. Hàm mất mát bao gồm điểm tương đồng có được từ mô-đun ZSL. Sau khi được đào tạo, mô hình phân loại một số với phần còn lại sau đó có thể dự đoán nhãn của một hình ảnh chưa biết bằng cách gán cho nó lớp mô tả văn bản có điểm tương đồng cao nhất. Ví dụ, nếu nhúng hình ảnh gần với nhúng văn bản có nội dung "một con sư tử", mô hình sẽ phân loại hình ảnh là một con sư tử.
- Các mô-đun nhúng ngữ nghĩa và trực quan là các mạng nơ-ron chiều hình ảnh và văn bản lên không gian nhúng. Các mô-đun có thể là các mô hình học sâu riêng biệt được đào tạo trên thông tin phụ trợ, như ImageNet. Đầu ra từ các mô hình này được đưa vào mô-đun ZSL và được đào tạo riêng biệt bằng cách giảm thiểu hàm mất mát độc lập. Ngoài ra, các mô-đun này có thể được đào tạo song song, như minh họa bên dưới.



Đào tạo chung các mô-đun ZSL

- Một trình trích xuất tính năng được đào tạo trước sẽ chuyển đổi hình ảnh con mèo trong hình minh họa ở trên thành một vector N chiều. Vector này biểu diễn các tính năng trực quan của hình ảnh được đưa vào mạng nơ-ron. Đầu ra của mạng nơ-ron là một vector tính năng chiều thấp hơn. Sau đó, mô hình sẽ so sánh vector tính năng chiều thấp hơn này với vector thuộc tính lớp đã biết và sử dụng truyền ngược để giảm thiểu tổn thất (sự khác biệt giữa cả hai vector).
- Tóm lại, khi bạn có được hình ảnh của một lớp mới, chưa biết (không phải là một phần của dữ liệu đào tạo), bạn sẽ:
 1. Trích xuất các tính năng bằng trình trích xuất tính năng.
 2. Chiếu các đặc điểm này vào không gian ngữ nghĩa bằng cách sử dụng mạng chiếu.

3. Tìm vector thuộc tính gần nhất trong không gian ngữ nghĩa để xác định lớp của hình ảnh.

2.5.4. Đánh giá các mô hình Zero-Shot Learning (ZSL)

Các học viên sử dụng một số số liệu đánh giá để xác định hiệu suất của các mô hình học tập zero-shot trong các tình huống thực tế. Các phương pháp phổ biến bao gồm:

- Độ chính xác Top-K: Chỉ số này đánh giá xem lớp thực tế có khớp với các lớp dự đoán với xác suất top-k hay không. Ví dụ, xác suất lớp có thể là 0,1, 0,2 và 0,15 đối với bài toán phân loại ba lớp. Với độ chính xác top-1, mô hình hoạt động tốt nếu lớp dự đoán có xác suất cao nhất (0,2) khớp với lớp thực tế. Với độ chính xác top-2, mô hình hoạt động tốt nếu lớp thực tế khớp với bất kỳ lớp nào trong số các lớp dự đoán với điểm xác suất top-2 là 0,2 và 0,15.
- Harmonic Mean: Bạn có thể tính toán harmonic mean—số giá trị chia cho nghịch đảo của trung bình số học—from các giá trị độ chính xác top-1 và top-5 để có kết quả cân bằng hơn. Nó giúp đánh giá hiệu suất mô hình trung bình bằng cách kết hợp độ chính xác top-1 và top-5.
- Diện tích dưới đường cong (AUC): AUC đo diện tích dưới đường cong đặc tính hoạt động của máy thu (ROC), tức là một biểu đồ cho thấy sự đánh đổi giữa tỷ lệ dương tính thực (TPR) hoặc thu hồi so với tỷ lệ dương tính giả (FPR) của một bộ phân loại. Bạn có thể đo hiệu suất phân loại tổng thể của mô hình ZSL dựa trên số liệu này.
- Độ chính xác trung bình trung bình (mAP): Số liệu mAP được sử dụng đặc biệt để đo độ chính xác của các tác vụ phát hiện đối tượng. Nó dựa trên việc đo độ chính xác và độ thu hồi cho mọi lớp nhất định ở nhiều mức ngưỡng tin cậy khác nhau. Phương pháp này giúp đo hiệu suất cho các tác vụ yêu cầu nhận dạng nhiều đối tượng trong một hình ảnh duy nhất. Nó cũng cho phép bạn xếp hạng điểm độ chính xác trung bình cho các ngưỡng khác nhau và xem ngưỡng nào mang lại kết quả tốt nhất.

2.6. CLIP của OpenAI

2.6.1. Giới thiệu về CLIP

CLIP (Contrastive Language–Image Pretraining) là một mô hình học sâu được phát triển bởi OpenAI, nhằm kết nối ngôn ngữ tự nhiên với hình ảnh. CLIP được thiết kế để hiểu và liên kết thông tin giữa hai phương thức dữ liệu này một cách hiệu quả, mở ra khả năng thực hiện các nhiệm vụ như phân loại hình ảnh, tìm kiếm hình ảnh dựa trên truy vấn văn bản, và học không giám sát (unsupervised learning).

2.6.2. Kiến trúc của CLIP

Mô hình CLIP bao gồm hai thành phần chính:

- Encoder Hình Ảnh: Sử dụng kiến trúc Vision Transformer (ViT) hoặc ResNet để trích xuất đặc trưng từ hình ảnh. Trong trường hợp của ViT-L/14, mô hình có khoảng 427 triệu parameters, cho phép xử lý và biểu diễn các đặc trưng phức tạp của hình ảnh.

- Encoder Văn Bản: Sử dụng kiến trúc Transformer tương tự như BERT để trích xuất đặc trưng từ văn bản. Thành phần này chuyển đổi các câu mô tả văn bản thành vector biểu diễn trong không gian ngữ nghĩa.

2.6.3. Cơ chế hoạt động của CLIP

CLIP được huấn luyện thông qua học tương phản (contrastive learning), mục tiêu là học cách ánh xạ hình ảnh và văn bản vào một không gian chung sao cho các cặp hình ảnh-văn bản liên quan nhau có vector biểu diễn gần nhau, trong khi các cặp không liên quan cách xa nhau.

Quá trình huấn luyện bao gồm các bước chính:

1. Chuẩn bị Dữ liệu: Sử dụng một tập hợp lớn các cặp hình ảnh và văn bản mô tả tương ứng.
2. Trích xuất Đặc trưng: Sử dụng encoder hình ảnh và encoder văn bản để chuyển đổi từng phần tử trong cặp dữ liệu thành vector biểu diễn.
3. Hàm Mất mát Tương phản: Áp dụng hàm mất mát tương phản nhằm tối ưu hóa mô hình sao cho các vector của cặp hình ảnh-văn bản liên quan được xếp gần nhau hơn trong không gian biểu diễn, trong khi các vector không liên quan được xếp xa nhau.
4. Tối ưu Hóa Mô hình: Sử dụng các thuật toán tối ưu như Adam để điều chỉnh các tham số của mô hình nhằm giảm thiểu hàm mất mát.

2.6.4. Lợi ích và Ứng dụng của CLIP

Lợi ích chính của CLIP bao gồm:

- Khả năng Zero-Shot Learning: CLIP có thể thực hiện các nhiệm vụ phân loại mà không cần được huấn luyện trước trên các lớp cụ thể. Điều này đạt được thông qua việc sử dụng các câu mô tả văn bản đại diện cho các lớp cần phân loại.
- Linh hoạt và Tổng quát: CLIP có khả năng áp dụng vào nhiều lĩnh vực khác nhau như tìm kiếm hình ảnh, phân loại đa lớp, và phân tích nội dung hình ảnh dựa trên ngôn ngữ tự nhiên.
- Hiệu quả Cao trong Học Liên kết Hình Ảnh-Văn Bản: CLIP cho phép hiểu sâu hơn về mối quan hệ giữa hình ảnh và văn bản, từ đó cải thiện khả năng tìm kiếm và trích xuất thông tin.

Các ứng dụng thực tiễn của CLIP:

- Phân loại Hình ảnh: Sử dụng CLIP để phân loại hình ảnh dựa trên các mô tả văn bản mà không cần huấn luyện lại mô hình cho từng lớp cụ thể.
- Tìm kiếm Hình ảnh Dựa trên Văn bản: Người dùng có thể truy vấn hình ảnh bằng cách sử dụng mô tả văn bản, CLIP sẽ trả về các hình ảnh phù hợp nhất.
- Phân tích và Trích xuất Thông tin Từ Video: Kết hợp với các kỹ thuật xử lý video khác, CLIP có thể giúp truy vấn và phân loại các sự kiện trong video dựa trên nội dung ngữ cảnh.

- Tạo Nội dung Đa phương tiện: CLIP có thể được sử dụng trong việc tạo ra các hệ thống đề xuất nội dung hoặc hỗ trợ trong việc tạo ra nội dung đa phương tiện dựa trên yêu cầu văn bản.

2.6.5. So sánh CLIP với Các Mô hình Khác

So với các mô hình truyền thống như CNN hay RNN, CLIP thể hiện những ưu điểm vượt trội về khả năng kết hợp thông tin giữa hình ảnh và ngôn ngữ. Trong khi CNN chỉ tập trung vào việc trích xuất đặc trưng hình ảnh và RNN xử lý chuỗi dữ liệu như văn bản, CLIP mang lại sự kết hợp mạnh mẽ giữa hai loại dữ liệu này thông qua kiến trúc Transformer hiện đại.

So sánh với OpenAI GPT:

- GPT là mô hình ngôn ngữ tự nhiên, chuyên về xử lý và tạo sinh văn bản.
- CLIP mở rộng khả năng này bằng cách kết hợp với hình ảnh, cho phép hiểu và liên kết thông tin giữa ngôn ngữ và thị giác.

So sánh với Google BERT:

- BERT tập trung vào việc hiểu ngữ cảnh trong văn bản.
- CLIP không chỉ hiểu ngữ cảnh văn bản mà còn kết hợp với hiểu biết về hình ảnh, tạo nên một mô hình đa phương thức.

2.6.6. Thách thức và Hạn chế của CLIP

Mặc dù CLIP mang lại nhiều ưu điểm, nhưng cũng tồn tại một số thách thức và hạn chế:

- Yêu cầu Tài nguyên Tính toán Cao: Với số lượng parameters lớn, việc huấn luyện và triển khai CLIP đòi hỏi phần cứng mạnh mẽ và nhiều tài nguyên tính toán.
- Khả năng Giải thích Hạn chế: Mặc dù CLIP mạnh mẽ trong việc kết nối hình ảnh và văn bản, nhưng việc giải thích quá trình ra quyết định của nó vẫn còn khó khăn.
- Phụ thuộc Vào Dữ liệu Đa dạng: Hiệu quả của CLIP phụ thuộc vào chất lượng và sự đa dạng của dữ liệu huấn luyện. Nếu dữ liệu không đa dạng, mô hình có thể gặp khó khăn trong việc khái quát hóa.

2.6.7. Zero-Shot Learning trong CLIP

Trong CLIP (Contrastive Language–Image Pretraining) của OpenAI, ZSL được tích hợp thông qua cách tiếp cận học tương phản giữa hình ảnh và văn bản:

- Đào tạo không giám sát: CLIP được huấn luyện trên một tập dữ liệu lớn gồm các cặp hình ảnh và văn bản liên kết mà không có nhãn cụ thể cho từng nhiệm vụ. Điều này cho phép mô hình học được cách liên kết giữa các đặc trưng hình ảnh và ngôn ngữ tự nhiên.
- Không gian biểu diễn chung: CLIP ánh xạ cả hình ảnh và văn bản vào một không gian biểu diễn chung, nơi các vector của hình ảnh và mô tả văn bản liên quan được đưa gần nhau, trong khi các vector không liên quan được cách xa nhau. Điều này cho phép CLIP thực hiện các nhiệm vụ như phân loại hình ảnh dựa trên mô tả văn bản mới mà không cần huấn luyện lại mô hình.
- Khả năng tổng quát hóa cao: Nhờ vào ZSL, CLIP có thể áp dụng kiến thức đã học được từ dữ liệu huấn luyện rộng lớn để giải quyết các nhiệm vụ mới, như phân loại

hình ảnh với các lớp chưa từng thấy trước đó, chỉ dựa trên mô tả văn bản của các lớp này.

2.6.8. Vision Transformer (ViT) trong CLIP

Cách thức hoạt động của ViT trong CLIP bao gồm:

- Chia nhỏ hình ảnh thành các patch: Hình ảnh đầu vào được chia thành các mảnh nhỏ (patches) có kích thước cố định, ví dụ 14x14 pixel. Mỗi patch được định dạng lại thành một vector và truyền qua một lớp chiếu tuyến tính để tạo thành các embedding.
- Thêm embedding vị trí: Tương tự như trong mô hình Transformer cho ngôn ngữ, các embedding vị trí được thêm vào các patch embedding để cung cấp thông tin về vị trí không gian của mỗi patch trong hình ảnh.
- Xử lý qua các lớp Transformer: Các patch embedding sau khi được thêm embedding vị trí sẽ được đưa qua nhiều lớp Transformer encoder, nơi chúng qua các cơ chế self-attention để học các mối quan hệ phức tạp giữa các patch.
- Tạo vector biểu diễn cuối cùng: Sau khi xử lý qua các lớp Transformer, vector biểu diễn của toàn bộ hình ảnh được lấy từ embedding của token [class] hoặc bằng cách áp dụng các kỹ thuật pooling. Vector này sẽ nằm trong cùng một không gian biểu diễn với vector văn bản, cho phép thực hiện các tác vụ tương phản giữa hình ảnh và văn bản.
- ViT trong CLIP cho phép mô hình học được các đặc trưng hình ảnh phức tạp và toàn cục, vượt qua những hạn chế của các mô hình CNN truyền thống trong việc bắt giữ thông tin không gian dài hạn.

2.6.9. Mạng Phần Dư (ResNet) trong CLIP

Cách thức hoạt động của ResNet trong CLIP bao gồm:

- Deep Residual Learning: ResNet được thiết kế với các khối residual, cho phép mô hình học các hàm đồng nhất (identity functions) để giải quyết vấn đề gradient vanishing trong các mạng sâu. Điều này giúp mô hình học được các đặc trưng phong phú từ hình ảnh.
- Trích xuất đặc trưng sâu: ResNet qua nhiều lớp tích chập và pooling để trích xuất các đặc trưng cấp cao từ hình ảnh, từ đó tạo ra vector biểu diễn cho mỗi hình ảnh.
- Tích hợp với CLIP: Trong CLIP, ResNet được sử dụng làm bộ mã hóa hình ảnh thay thế cho ViT. Vector biểu diễn của hình ảnh từ ResNet được ánh xạ vào không gian biểu diễn chung cùng với vector văn bản từ bộ mã hóa ngôn ngữ, cho phép thực hiện các tác vụ tương phản và Zero-Shot Learning tương tự như khi sử dụng ViT.

So sánh giữa ViT và ResNet trong CLIP:

- ViT:
 - Tốt hơn trong việc nắm bắt mối quan hệ toàn cục giữa các phần của hình ảnh nhờ cơ chế self-attention.
 - Yêu cầu dữ liệu huấn luyện lớn để đạt hiệu quả tối ưu.
 - Có khả năng mở rộng linh hoạt với quy mô mô hình.
- ResNet:
 - Tốt trong việc trích xuất đặc trưng cục bộ và là kiến trúc đã được chứng minh hiệu quả trên nhiều nhiệm vụ thị giác.

- Thường hiệu quả hơn trên các tập dữ liệu nhỏ hoặc trung bình.
- Ít yêu cầu tài nguyên tính toán hơn so với ViT khi xử lý các hình ảnh phức tạp.

CHƯƠNG 3: XÂY DỰNG ỨNG DỤNG VÀ KẾT QUẢ THỰC NGHIỆM

CHƯƠNG 4: ĐỊNH HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI

TÀI LIỆU THAM KHẢO