Phương pháp deep learning với mạng Nơ ron tích chập

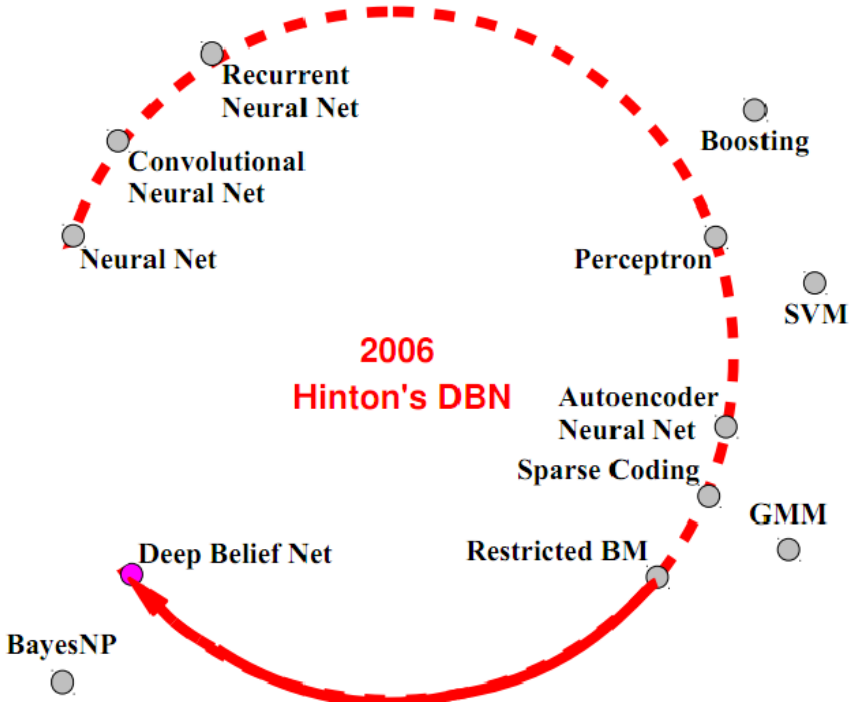
1 Khái niệm về deep learning và lịch sử phát triển của nó

Deep learning là sự đột phá trong việc nghiên cứu về mạng Nơ ron, phương pháp học này đặc biệt được quan tâm từ sau công trình nghiên cứu của Hilton năm 2006. Sự đột phá ở đây gồm một số khía cạnh sau:

- Số lượng lớp của mạng tăng lên đáng kể: có thể đến vài chục hoặc thậm chí hàng trăm lớp ẩn.
- Số lượng lớp tăng làm cho số lượng tham số cũng tăng lên rất nhiều đồ hỏi phải có thuật toán học hiệu quả.
- Kích thước mạng lớn đòi hỏi phải có các hệ thống máy tính tốc độ cao để xử lý. Trong những năm gần đây sự phát triển của GPU đã trợ giúp cho quá trình này.
- Số lượng dữ liệu dùng cho huấn luyện lớn. Điều này trước đây là khó thực hiện tuy nhiên đối với các hệ thống học sâu chúng ta phải có số lượng dữ liệu mẫu đủ lớn.

Deep learning là một nhánh của lĩnh vực học máy, dựa trên một tập hợp các thuật toán nhằm cố gắng mô hình dữ liệu trừu tượng hóa ở mức cao bằng cách sử dụng nhiều lớp xử lý với cấu trúc phức tạp, hoặc bao gồm nhiều biến đổi phi tuyến.

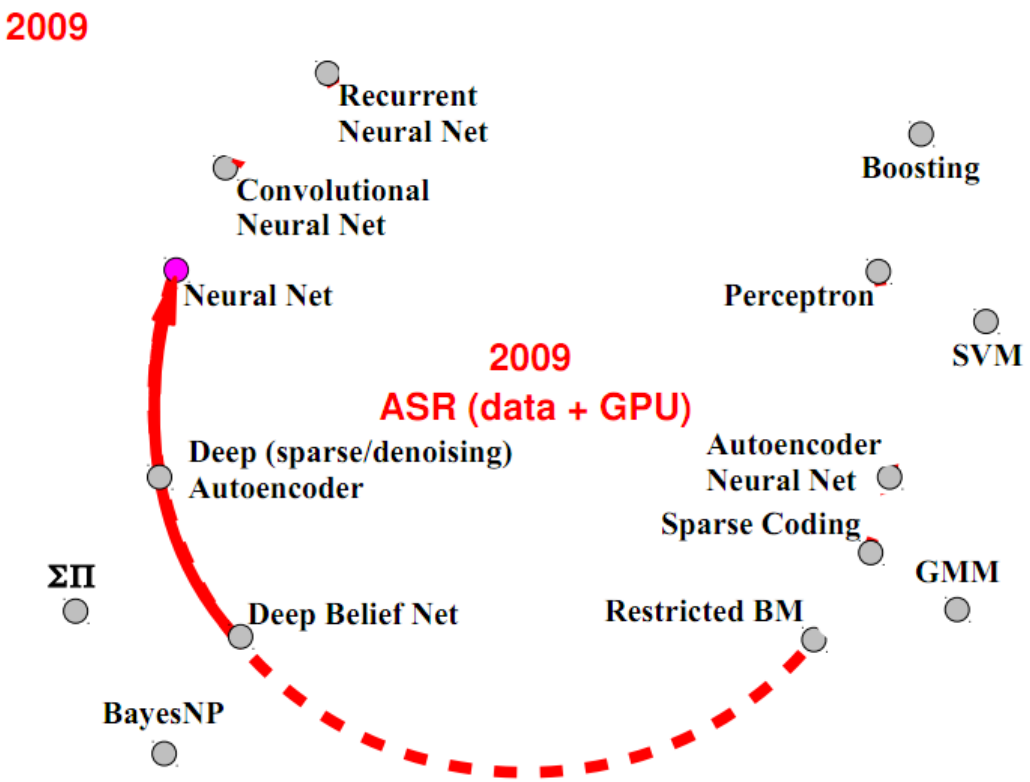
Hai loại mạng Nơ ron được sử dụng nhiều nhất trong thực tế là mạng CNN và RNN. Mạng RNN phù hợp với các bài toán trong xử lý ngôn nhữ tự nhiên trong khi đó mạng CNN phù hợp với các bài toán về xử lý ảnh, nhận dạng đối tượng,…



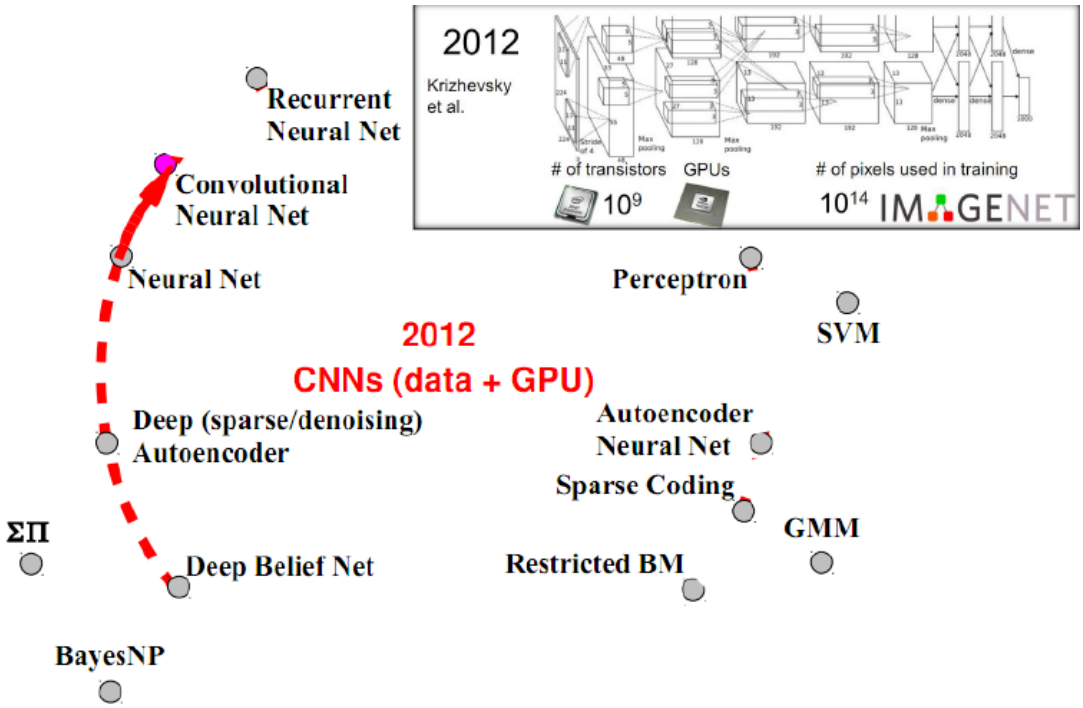
Hình 2.13 Mốc đánh dấu sự ra đời của deep learning năm 2006

Năm 2006 được coi là một dấu mốc quan trong đánh dấu sự đột phá vượt bậc trong nghiên cứu về mạng Nơ ron bởi công trình nghiên cứu của Hilton về mạng nơ ron Niềm tin sâu (Deep Belief Net) (xem hình 2.13). Ngày sau công trình này các nghiên cứu về deep learning trở nên vô cùng mạnh mẽ. Hình 2.14 minh họa các nghiên cứu đến năm 2009 về deep learning.

Tiếp theo là nghiên cứu về học sâu với hai loại mạng quan trọng là mạng Nơ ron tích chập và mạng Nơ ron hồi quy. Trong đó mạng Nơ ron tích chập được thiết kế với rất nhiều ưu điểm cho lớp bài toán xử lý và khai phá dữ liệu đối với ảnh (hình 2.15).



Hình 2.14 Các nghiên cứu tiếp sau sự ra đời của deep learning



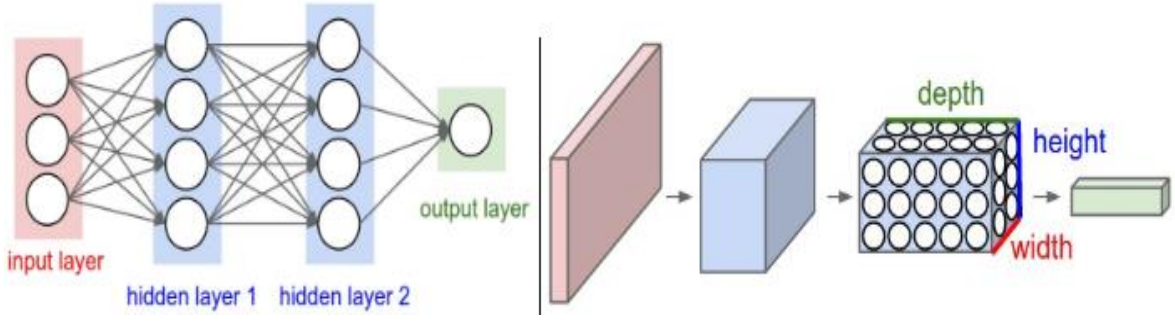
Hình 2.15 Các nghiên cứu về deep learning đến nay

Cũng trong thời gian này hàng loạt các công ty công nghệ đã tập trung nghiên cứu và ứng dụng deep learning; chúng ta có thể kể đến như Google, Faceboook, Baidu, IBM,... Những ứng dụng đã triển khai ra thực tế rất quan trọng như hệ thống xe tự lái, hệ thống chatbot, hệ thống nhận dạng đối tượng,...

Tóm lại điều làm cho học đa tầng khác biệt là (1) chất lượng vượt trội đối với một số lĩnh vực cần giải quyết, (2) làm cho việc giải quyết vấn đề đơn giản hơn vì gần như các công đoạn được tự động điều chính nhất là khâu trích chọn đặc trưng. Hai tính chất quan trọng có thể giải thích thêm cho việc nổi trội về chất lượng của các hệ thống deep learning là quá trình học được bổ sung dần, từ lớp này đến lớp khác trong đó các mô tả về đối tượng có độ phức tạp tăng dần và quá trình học được liên kết với nhau từ đầu đến cuối.

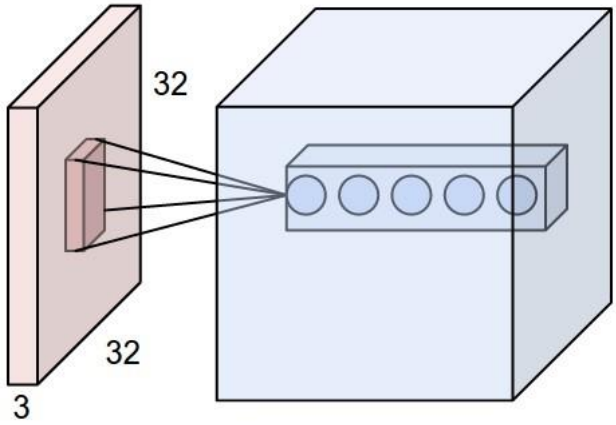
2 Học đa tầng với mô hình mạng Nơ ron tích chập

Đầu tiên, với mạng Nơ ron tích chập (Convolutional Neural Network) hay CNN, đây là một kiến trúc mạng Nơ ron nhiều lớp. Điểm khác biệt nhất của mạng Nơ ron tích chập so với mạng Nơ ron thông thường là ở mỗi lớp gồm nhiều tầng xếp chồng lên nhau và có độ sâu nào đó (xem hình 2.16).



Hình 2.16 Mạng Nơ ron (trái) và mạng Nơ ron tích chập (phải)

Chúng tôi cũng lưu ý rằng mạng Nơ ron tích chập đặc biệt phù hợp cho dữ liệu đầu vào có dạng ma trận chẳng hạn ma trận biểu diễn cho ảnh.



Hình 2.17 Minh họa quá trình kết nối địa phương trong mạng CNN

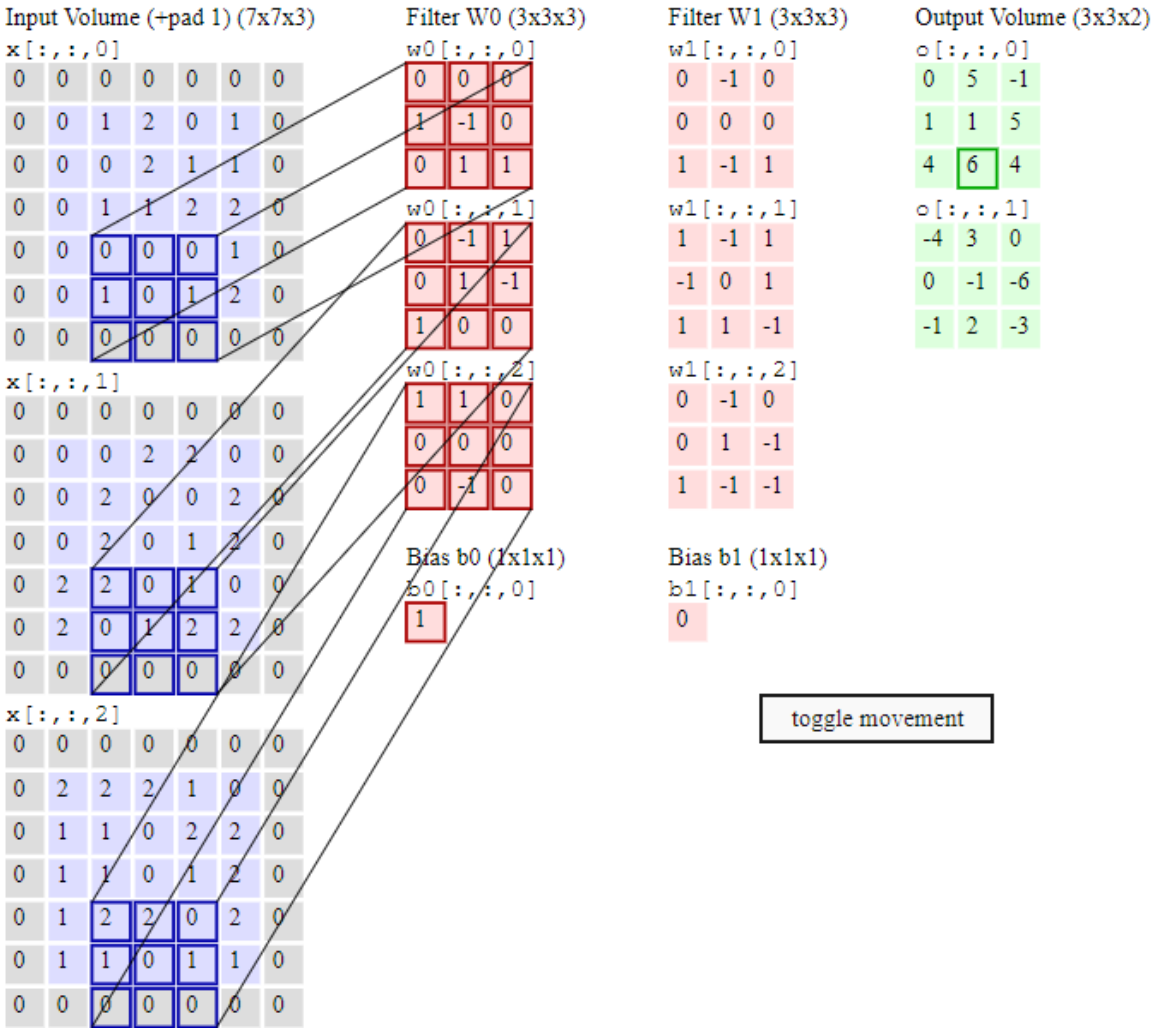
Một số khái niệm cơ bản về mạng CNN được trình bày như sau [8, 9, 15, 16]:

Lớp tích chập (convolutional layer)

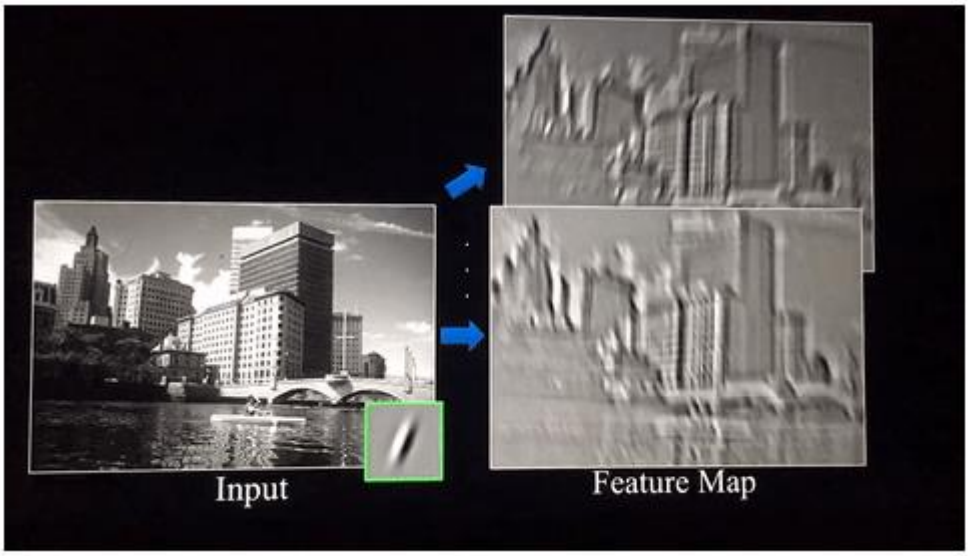
Với CNN, lớp tích chập cũng chính là các lớp ẩn, khác ở chỗ, lớp tích chập gồm các *feature map***,** về bản chất mỗi *feature map* này là một bản quét qua của các đầu vào ban đầu nhưng được trích xuất ra các đặc tính cụ thể. Quá trình quét như thế nào thì lại dựa vào các bộ lọc (*Convolution Filter***)** (hình 2.18)**.** Các bộ lọc tích chập sẽ thực hiện việc quét qua ma trận dữ liệu đầu vào, từ trái qua phải, trên xuống dưới, và nhân tương ứng từng giá trị của ma trận đầu vào mà ma trận lọc rồi cộng tổng lại, đưa qua các hàm kích hoạt (sigmoid, relu, tanh, ...), kết quả sẽ là một con số cụ thể, tập hợp các con số này lại là 1 ma trận nữa, chính là đặc trưng.

Với ví dụ hình 2.18 dữ liệu đầu vào có kích thước 5×5×3, sau khi làm hàng rào bên ngoài ta sẽ có kích thước là 7×7×3. Để đi xây dựng các tầng cho các lớp ta phải sử dụng các bộ lọc (số lượng bộ lọc tùy theo thực tế). Chẳng hạn với vị dụ này ta xét 2 bộ lọc W0 và W1. Với đầu vào như vậy thì sẽ có 2 tầng ở lớp tiếp theo sau khi tính toán. Mỗi giá trị của một tầng sẽ được tính toán qua 3 vùng giá trị của ma trận đầu vào tương ứng, trong ví dụ đang xét cho giá trị 6 ở tầng số 1.

Sau khi tính toán xong thì lớp đầu ra sẽ gồm 2 tầng mỗi tầng có kích thước 3×3 và kích thước của lớp này sẽ là 3×3×2 tương ứng là chiều rộng (width), chiều cao (height) và chiều sâu (depth) của lớp đó. Như vậy nếu ta sử dụng 4 bộ lọc thì ta sẽ có kích thước khối đầu ra gồm các nơ ron là 3×3×4. Hình 2.19 minh họa ví dụ về các *feature map*.



Hình 2.18 Ví dụ về việc tính toán các lớp cho mạng Nơ ron tích chập



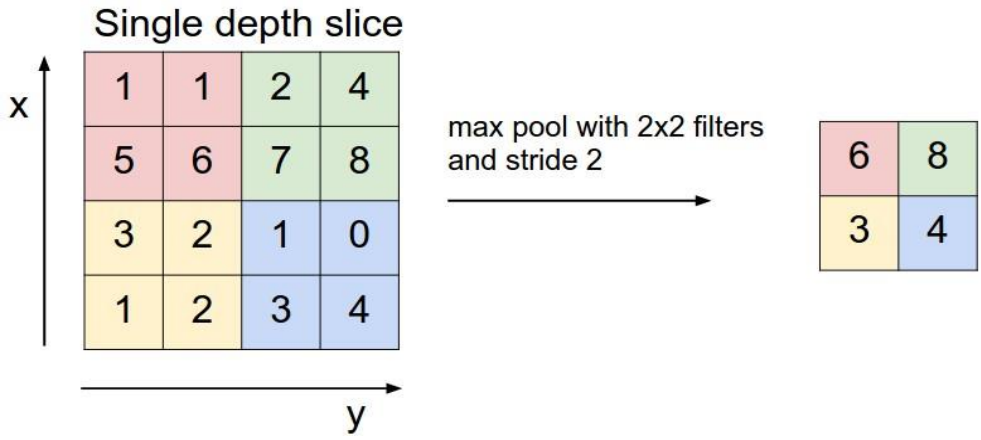
Hình 2.19 Ví dụ về quá trình sinh ra các đặc trưng cho một ảnh

Stride and Padding

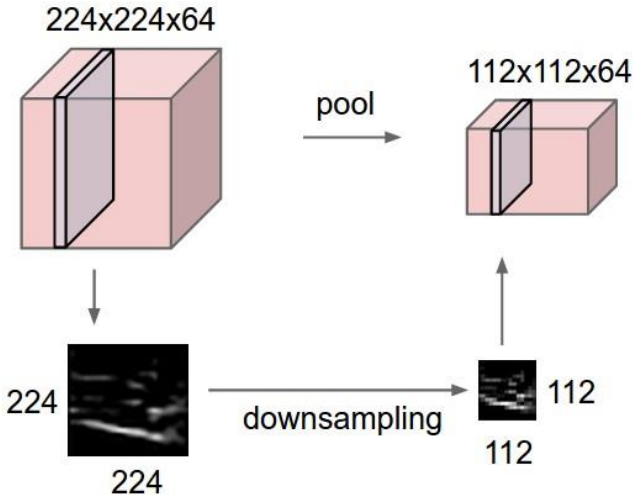
Stride là khoảng cách giữa 2 bộ lọc khi quét. Với stride = 1 thì sẽ quét 2 ô ngay cạnh nhau, nhưng với stride = 2, quá trình sẽ quét ô số 1, số 3, số 5, Padding là việc lập hàng rào bao quanh ma trận dữ liệu với các số 0 để quá trình sinh ra các feature map không bị mất mát thông tin bởi các ô ở rìa biên.

Pooling

Mục đích của quá trình này là làm giảm kích thước của dữ liệu. Khác với lớp Convolutional, lớp Pooling không tính tích chập mà tiến hành lấy mẫu (subsampling). Khi cửa sổ trượt trên ảnh, chỉ có một giá trị được xem là giá trị đại diện cho thông tin ảnh tại vùng đó (giá trị mẫu) được giữ lại. Các phương thức lấy phổ biến trong lớp Pooling là MaxPooling (lấy giá trị lớn nhất), MinPooling (lấy giá trị nhỏ nhất) và AveragePooling (lấy giá trị trung bình). Hình 2.20 và 2.21 minh họa ví dụ về max-pooling.



Hình 2.20 Ví dụ về sử dụng max-pooling



Hình 2.21 Ví dụ về việc giảm kích thước của dữ liệu: từ một khối kích thước 224×224×64 sau khí giảm chúng ta sẽ được khối 112×112×64

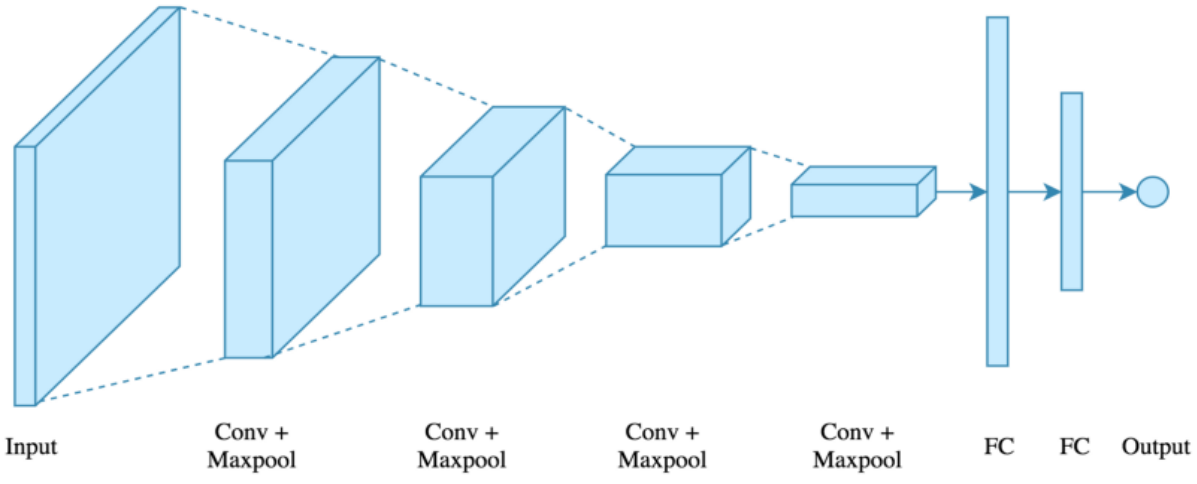
Trong những năm gần đây đối với mạng Nơ ron tích chập, hàm kích hoạt hay sử dụng là hàm ReLU. Sau mỗi lần tính toán các lớp, để giảm kích thước của quá trình tính toán chúng ta sử dụng phương pháp giảm số chiều (max-pooling). Phương pháp sử dụng max-pooling rất đơn giản. Đôi khi sử dụng một ma trận quét qua từng tầng của lớp và lấy max của chúng.

Fully Connected

Thông thường thì sau các lớp Conv+Pooling thì sẽ là 2 lớp Fully connected, 1 layer để tập hợp các feature layer mà ta đã tìm ra, chuyển đổi dữ liệu từ 3D, hoặc 2D thành 1D, tức chỉ còn là 1 vector. Còn 1 layer nữa là output, số neuron của layer này phụ thuộc vào số output mà ta muốn tìm ra. Giả sử với tập dữ liêu MNIST chẳng hạn, ta có tập các số viết tay từ 0 -> 9. Vậy output sẽ có số neuron là 10, tập dữ liệu phân loại cảm xúc khuôn mặt có 7 loại thì số lượng nơ ron sẽ là 7, bài toàn xác định ảnh khuôn mặt là cười hay không thì chỉ cần 2 nơ ron.

Cấu trúc mạng Nơ ron

Cấu trúc chung của mạng Nơ ron tích chập được trình bày trong hình 2.22. Từ dữ liệu đầu vào ta lần lượt xây dựng các lớp, thực hiện max-pooling và hai lớp cuối cùng của mạng sẽ là các lớp kết nối đầy đủ (fully connected – FC). Thường thì sau các lớp Conv+Pooling thì sẽ là 2 lớp Fully connected, 1 layer để tập hợp các feature layer mà ta đã tìm ra, chuyển đổi dữ liệu từ 3D, hoặc 2D thành 1D, tức chỉ còn là 1 vector. Còn 1 layer nữa là output, số neuron của layer này phụ thuộc vào số output mà ta muốn tìm ra. Giả sử với tập dữ liêu MNIST chẳng hạn, ta có tập các số viết tay từ 0 -> 9. Vậy output sẽ có số neuron là 10.



Hình 2.22

Tóm lại cấu trúc của mạng deep learning CNN tổng quát như sau:

INPUT→[[CONV→RELU]*N→POOL?]*M→[FC→RELU]*K→FC

trong đó kí hiệu * thể hiện số lần lặp lại, POOL? thể hiện tùy chọn có/không, thông thường: $0 \leq N \leq 3$, $M \geq 0$, $0 \leq K \leq 3$.

Quá trình huấn luyện mạng deep learning CNN được mô tả trong hình 2.33.

Bước 1: Khởi tạo các giá trị tham số, và bộ lọc ngẫu nhiên.

Bước 2: Với mỗi ảnh huấn luyện, thực hiện các phép biến đổi sử dụng cho các bộ lọc để xác định lớp tích chập, sử dụng hàm kích hoạt, pooling, qua lớp FC và tính xác suất đầu ra tương ứng cho mỗi lớp. Với ảnh đầu tiên các giá trị là ngẫu nhiên nên kết quả đầu ra cũng sẽ ngẫu nhiên.

Bước 3: Tính giá trị lỗi tại lớp đầu ra.

Bước 4: Sử dụng thuật toán lan truyền ngược ước lượng các tham số theo phương pháp gradient. Lúc này giá trị các tham số của mạng CNN sẽ được cập nhật. Chú yes rằng các tham số như số lượng bộ lọc, kích thước bộ lọc, kiến trúc của mạng CNN đã được xác định từ đầu nên không thay đổi trong quá trình huấn luyện. Chỉ có các giá trị của bộ lọc và các trọng số liên kết giữa các nơ ron là sẽ được cập nhật.

Bước 5: Lặp lại bước 2-4 với tất cả các ảnh còn lại của tập dữ liệu huấn luyện.

Hình 2.23 Quá trình huấn luyện mạng CNN

Trong vài năm trở lại đây, liên tục có các cuộc thi liên quan đến đề xuất của các hệ thống deep learning, chúng tôi có thể điểm một số hệ thống chủ yếu sau đây:

LeNet: Các ứng dụng thành công đầu tiên của Mạng lưới chuyển tải được Yann LeCun phát triển vào năm 1990. Trong số này, nổi tiếng nhất là kiến trúc LeNet được sử dụng để đọc mã zip, chữ số, v.v. Trong nghiên cứu này Lecun đã công bố những mạng Nơ ron tích chập có số lớp là 5.

AlexNet: Công trình đầu tiên phổ biến Mạng lưới liên kết trong Mạng máy tính là AlexNet, được phát triển bởi Alex Krizhevsky, Ilya Sutskever và Geoff Hinton. AlexNet đã được gửi đến thách thức ImageNet ILSVRC trong năm 2012 và vượt trội hơn đáng kể so với đề xuất thứ hai (lỗi 5 hàng đầu 16% so với vị trí thứ hai 26%). Mạng có kiến trúc rất giống với LeNet, nhưng là các lớp Convolutional sâu hơn, lớn hơn và nổi bật xếp chồng lên nhau (trước đây nó thường chỉ có một lớp CONV duy nhất luôn được theo sau bởi một lớp POOL).

ZF Net: Đề xuất hệ thống deep learning đạt chất lượng tốt nhất tại cuộc thi ILSVRC 2013 là của nhóm Matthew Zeiler và Rob Fergus. Nó được gọi là ZFNet với viết tắt của Zeiler & Fergus Net. Đó là một sự cải tiến trên AlexNet bằng cách tinh chỉnh các tham số và kiến trúc, đặc biệt bằng cách mở rộng kích thước của các lớp trung gian.

GoogLeNet: Người chiến thắng ILSVRC 2014 là một giải pháp của nhóm Szegedy et al. từ Google. Đóng góp chính của nó là phát triển mô-đun khởi động làm giảm đáng kể số lượng tham số trong mạng (4M, so với AlexNet với 60M). Ngoài ra nghiên cứu này sử dụng tính năng Pool Pooling thay vì các lớp Fully Connected ở phía trên cùng của ConvNet, loại bỏ một lượng lớn các tham số mà dường như không quan trọng lắm. Ngoài ra còn có một số phiên bản tiếp theo cho GoogLeNet, bản gần đây nhất là Inception-v4.

VGGNet: Đứng thứ hai trong cuộc thi ILSVRC 2014 là đề xuất đến từ nhóm Karen Simonyan và Andrew Zisserman được biết đến với tên VGGNet. Đóng góp chính của nghiên cứu là cho thấy rằng chiều sâu của mạng là một thành phần quan

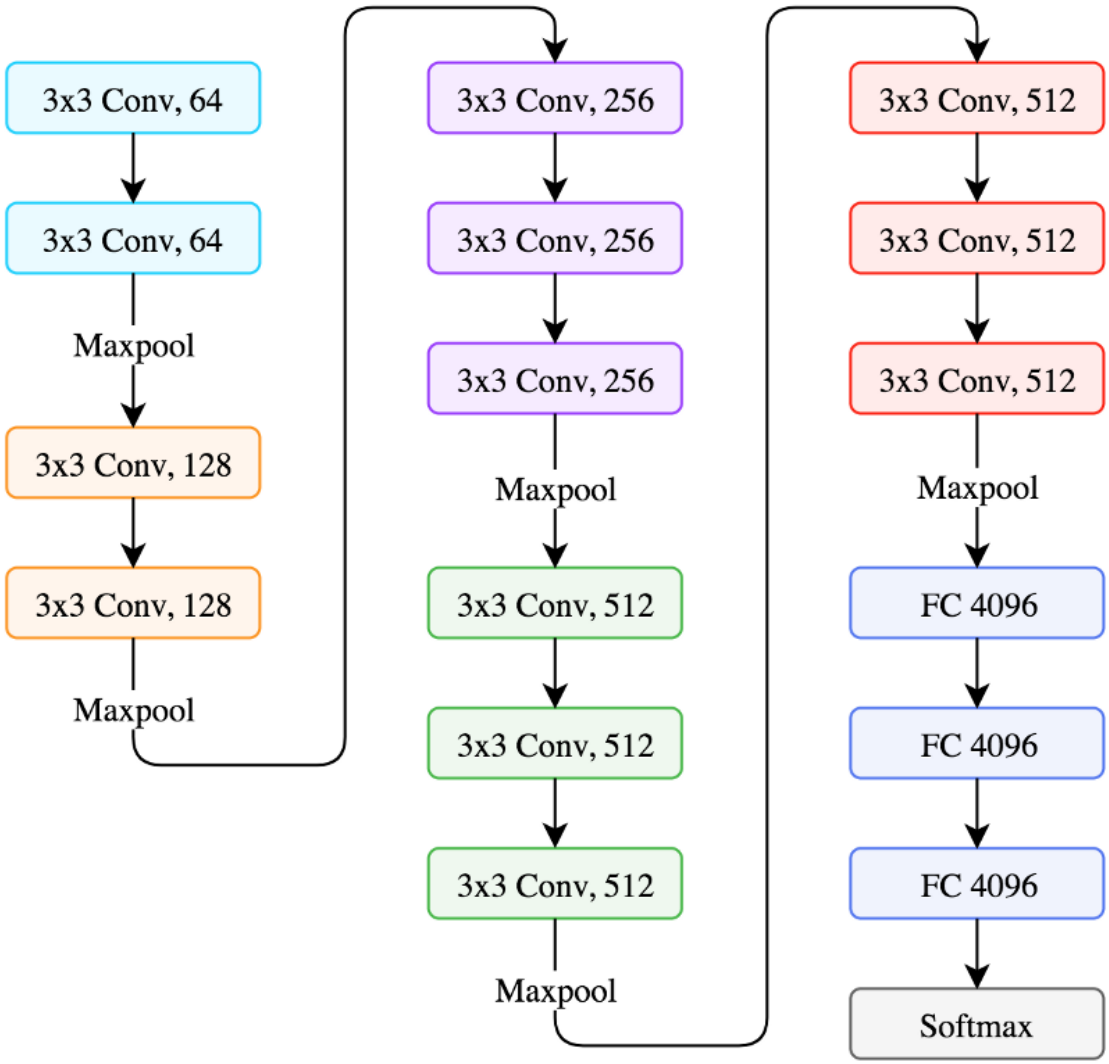
trọng nhằm cho hiệu suất tốt. Mạng đề xuất chứa 16 lớp CONV/ FC. Nhược điểm của VGGNet là nó kích thước lớn, với nhiều bộ nhớ và thông số hơn (140M) (xem hình 2.24.

Layer id	Layer type	Parameters number
(0):	Image (3, 224, 224)	
(1):	Convolution (64, 3x3x3, 1x1, 1x1)	1,792
(3):	Convolution (64, 64x3x3, 1x1, 1x1)	36,928
(5):	MaxPooling (2x2,2x2)	
(6):	Convolution (128, 64x3x3, 1x1, 1x1)	73,856
(8):	Convolution (128, 128x3x3, 1x1, 1x1)	147,584
(10):	MaxPooling (2x2,2x2)	
(11):	Convolution (256, 128x3x3, 1x1, 1x1)	295,168
(13):	Convolution (256, 256x3x3, 1x1, 1x1)	590,080
(15):	Convolution (256, 256x3x3, 1x1, 1x1)	590,080
(17):	MaxPooling (2x2,2x2)	
(18):	Convolution (512, 256x3x3, 1x1, 1x1)	1,180,160
(20):	Convolution (512, 512x3x3, 1x1, 1x1)	2,359,808
(22):	Convolution (512, 512x3x3, 1x1, 1x1)	2,359,808
(24):	MaxPooling (2x2,2x2)	
(25):	Convolution (512, 512x3x3, 1x1, 1x1)	2,359,808
(27):	Convolution (512, 512x3x3, 1x1, 1x1)	2,359,808
(29):	Convolution (512, 512x3x3, 1x1, 1x1)	2,359,808
(31):	MaxPooling (2x2,2x2)	
(32):	FullyConnected (25088 → 4096)	102,764,544
(34):	Dropout (50%)	
(35):	FullyConnected (4096 → 4096)	16,781,312
(37):	Dropout (50%)	
(38):	FullyConnected (4096 → 1000)	4,097,000
(40):	SoftMax	
Total :	**16 layers**	**138,357,544**

Hình 2.24 Số lượng tham số cho mạng VGG với 16 lớp

ResNet: Mạng còn lại được phát triển bởi Kaiming He et al. là nhóm chiến thắng của ILSVRC 2015. Mạng này được giới thiệu năm 2016 với kiến trúc rất nhiều lớp như sau:

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9



Hình 2.25 Kiến trúc mạng VGG