Convolutional Neural Networks

1. Convolutional Neural Network là gì?

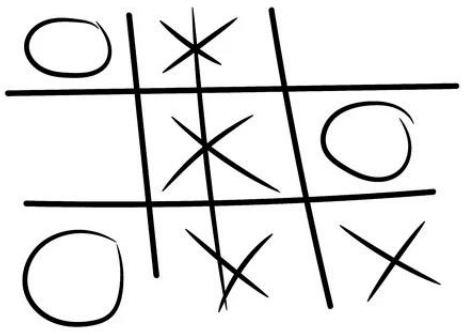
CNNs là một ANN được dùng để nhận dạng, phân loại ảnh. CNNs lần đầu được giới thiệu bởi Yann LeCun vào những năm 1980

LeNet là một trong những cấu trúc CNNs lâu đời nổi tiếng nhất phát triển vào những năm 1998s và đã có thể nhận dạng được chữ viết tay (accurency lên đến 90%)

2. Vậy tại sao cần đến Convolutional Neural Network?

Để hiểu sự cần thiết của CNNs, các bạn hãy thử hình dung bài toán sau:

Bạn đang chơi trò tick tack toe với crush của mình qua màn hình điện thoại, cách chơi rất đơn giản, chỉ cần vẽ O hoặc X lên một ma trận có 3x3 ô, ai tạo được một đường thẳng là người chiến thắng



Để giải quyết bài toán trên, bạn phải làm cho máy tính nhận dạng và hiểu được hình bạn vẽ là O hay X.

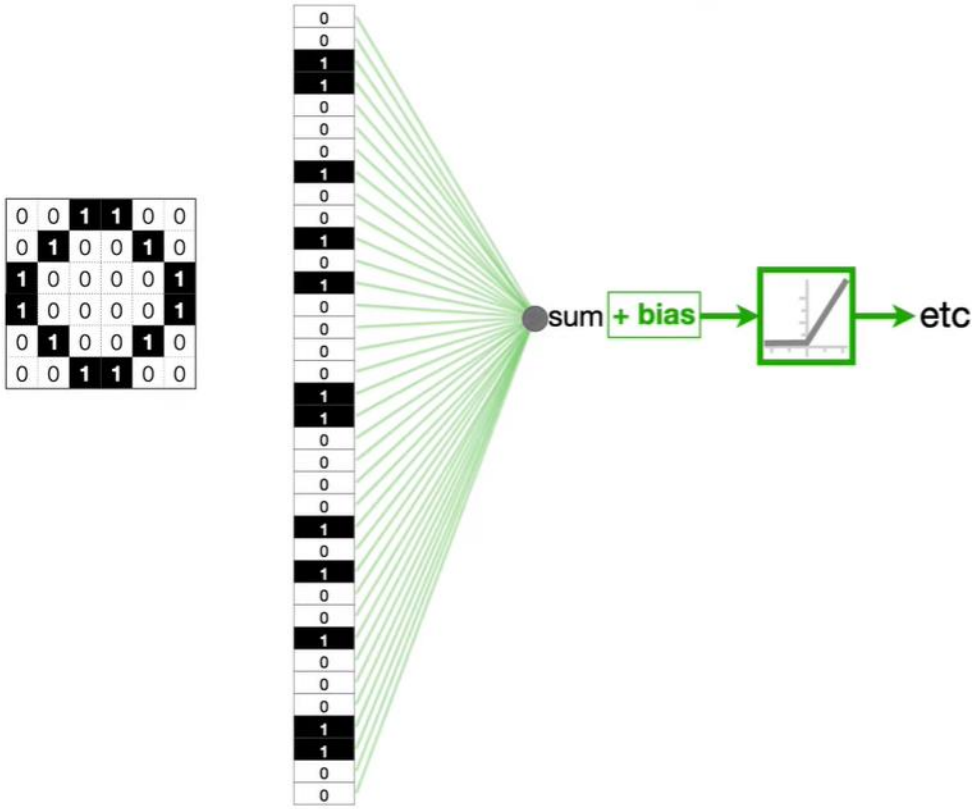
Thực ra hình ảnh chính là một ma trận 2 chiều, mỗi pixel của hình ảnh cũng có thể được biểu diễn mới một con số, bạn hãy quan sát hình bên dưới. Con số 0 ở

mỗi ô trên ma trận thể hiện rằng pixel của tấm ảnh đó là màu trắng và ngược lại, số 1 thể hiện pixel trên tấm ảnh là màu đen

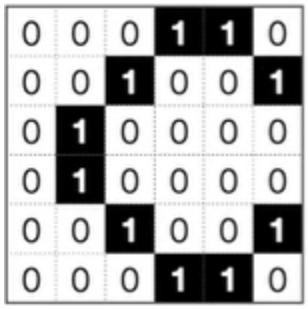
0	0	1	1	0	0
0	1	0	0	1	0
1	0	0	0	0	1
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
0	0	1	1	0	0
0	1	0	0	1	0
1	0	0	0	0	1

Hình ảnh trên có 6x6 pixel, vì vậy ta có thể chuyển nó thành 1 vector có 36 phần tử và đưa vector này làm input node cho hidden layer

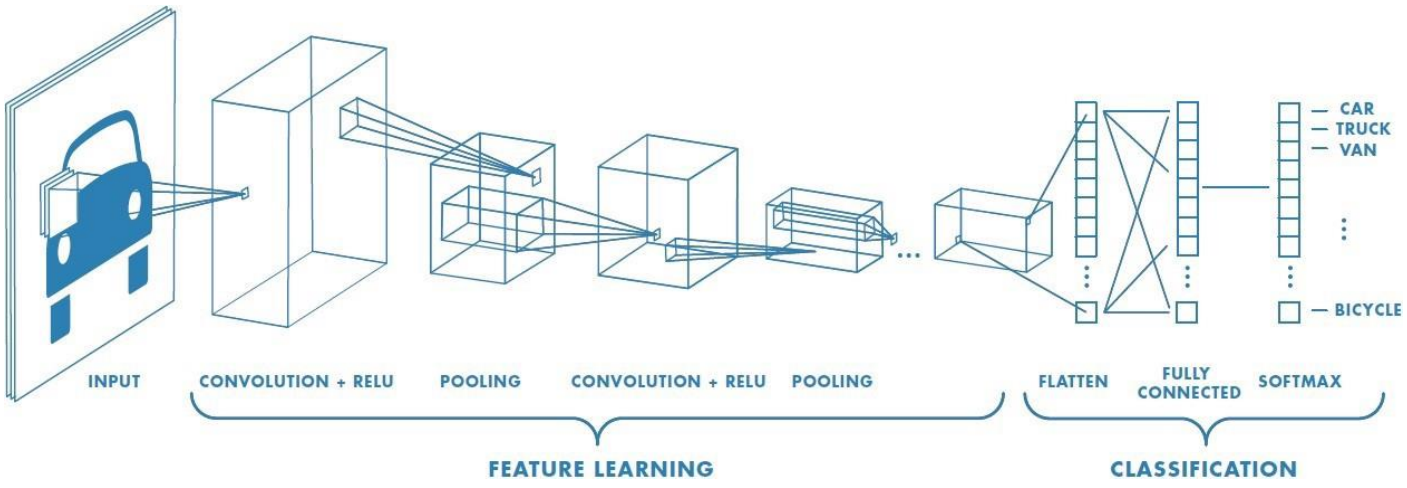


Lúc này ta có tất cả 36 cạnh kết nối từ 36 phần tử của vector đến hidden layer, có nghĩa là ta phải tính toán được 36 weights thông qua backpropagation. Nhưng thông thường hidden layer sẽ có nhiều hơn 1 node. Vậy nếu tấm ảnh đầu vào của chúng ta có kích thước lớn hơn ví dụ như 100x100 thì số lượng weights cần phải tính toán là 10.000 đối với mỗi node trong hidden layer, nên phương pháp này yêu cần tài nguyên rất lớn đối với những tấm ảnh thực tế. Và trong bài toán lúc đầu nếu như tấm ảnh của chúng ta bị xê dịch như hình dưới thì có vẻ ANN là một lựa chọn không được tốt.



Để khắc phục những bất cập trên của ANN, ta mới cần đến CNNs trong những bài toán xử lí ảnh.

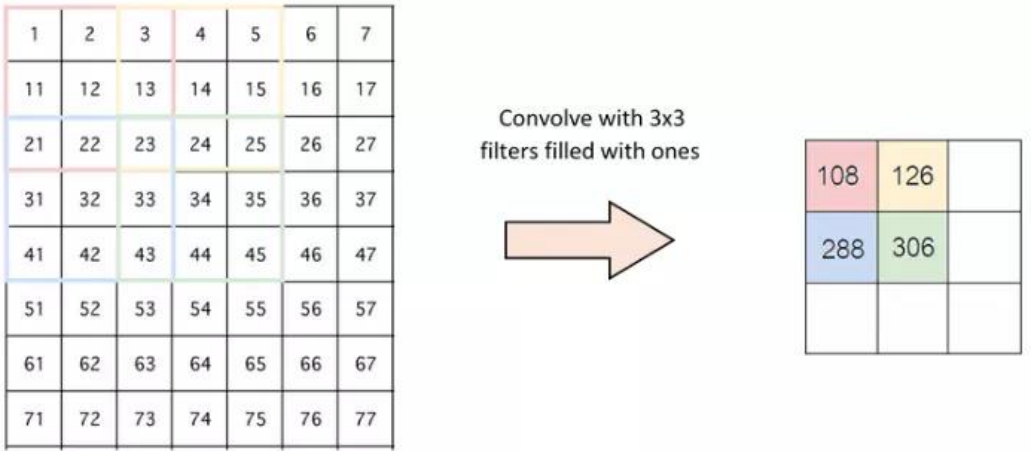
3. Các lớp cơ bản của mạng CNNs



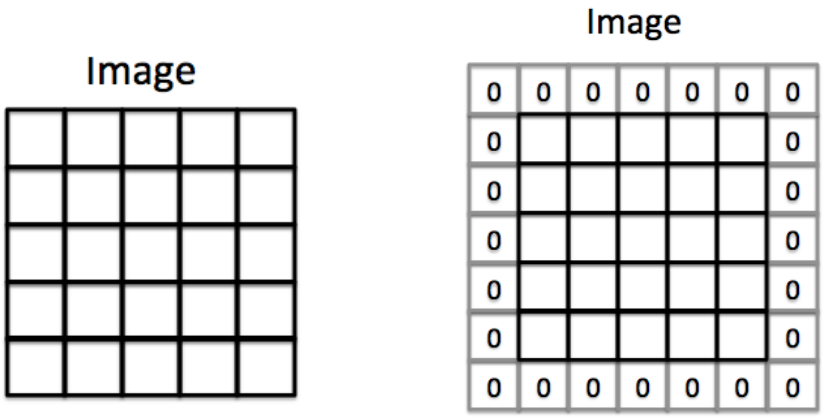
Convolutional layer

Đây là lớp quan trọng nhất của CNN, lớp này có nhiệm vụ thực hiện mọi tính toán. Những yếu tố quan trọng của một convolutional layer là: stride, padding, filter map, feature map.

Stride: là số pixel thay đổi trên ma trận đầu vào. Khi stride là 1 thì ta di chuyển các kernel 1 pixel. Khi stride là 2 thì ta di chuyển các kernel đi 2 pixel và tiếp tục như vậy. Hình dưới là lớp tích chập hoạt động với stride là 2



Padding: dùng để thêm các pixel bên ngoài hình ảnh, và Zero padding là mọi giá trị thêm bằng 0. Mục đích của padding là sau mỗi lần sử dụng các bộ lọc để quét ảnh, kích thước của ảnh sẽ nhỏ và nhỏ hơn nữa và sẽ không giữ nguyên kích thước ban đầu của ảnh và sẽ không thể khai thác được ảnh nữa, do đó cần thêm một số pixel bên ngoài vào hình ảnh



nếu zero padding=1 sẽ có một pixel dày xung quanh ảnh gốc với giá trị pixel=0

Feature map: Nó thể hiện kết quả của mỗi lần filter map quét qua input. Sau mỗi lần quét sẽ xảy ra quá trình tính toán

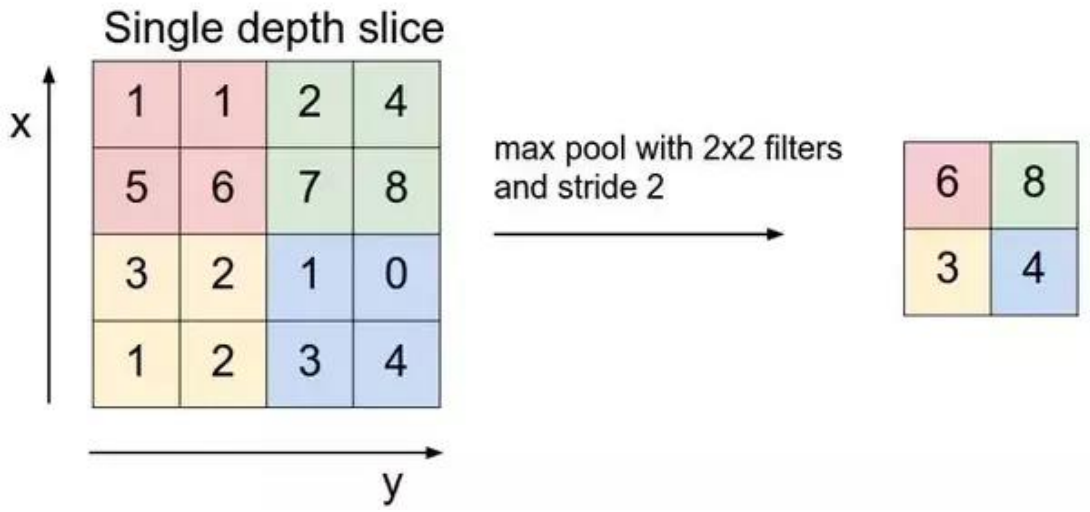
Relu

Relu là hàm kích hoạt trong neural network và hàm này còn được gọi là activation function. Hàm kích hoạt có tác dụng mô phỏng các neuron có tỷ lệ truyền xung qua axon. Trong activation function thì nó còn có hàm nghĩa là: Relu, Leaky, Tanh, Sigmoid, Maxout,...Hiện nay, hàm relu được dùng phổ biến và vô cùng thông dụng.

Nó được sử dụng nhiều cho các nhu cầu huấn luyện mạng neural thì Relu mang lại rất nhiều ưu điểm nổi bật như: việc tính toán sẽ trở nên nhanh hơn,... Quá trình sử dụng relu, chúng ta cần lưu ý đến vấn đề tùy chỉnh các learning rate và theo dõi dead unit. Những lớp relu layer đã được sử dụng sau khi filter map được tính ra và áp dụng hàm relu lên những giá trị của filter map.

Pooling layer

Lớp pooling thường được sử dụng ngay sau lớp convulational để đơn giản hóa thông tin đầu ra để giảm bớt số lượng neural. Các loại pooling phổ biến là max pooling và average pooling



Fully connected layer

Lớp này có nhiệm vụ đưa ra kết quả sau khi lớp convolutional layer và pooling layer đã nhận được ảnh truyền. Lúc này, ta thu được kết quả là model đã đọc được thông tin của ảnh và để liên kết chúng cũng như cho ra nhiều output hơn thì ta sử dụng fully connected layer.

Ngoài ra, nếu như fully connected layer có được giữ liệu hình ảnh thì chúng sẽ chuyển nó thành mục chưa được phân chia chất lượng. Cái này khá giống với phiếu bầu rồi chúng sẽ đánh giá để bầu chọn ra hình ảnh có chất lượng cao nhất.

4. CNNs hoạt động như thế nào?

What will be the output size, if the input to convolution layer of neural network is an image of size 128X128X3 and 40 filters of size 5X5 are applied to it

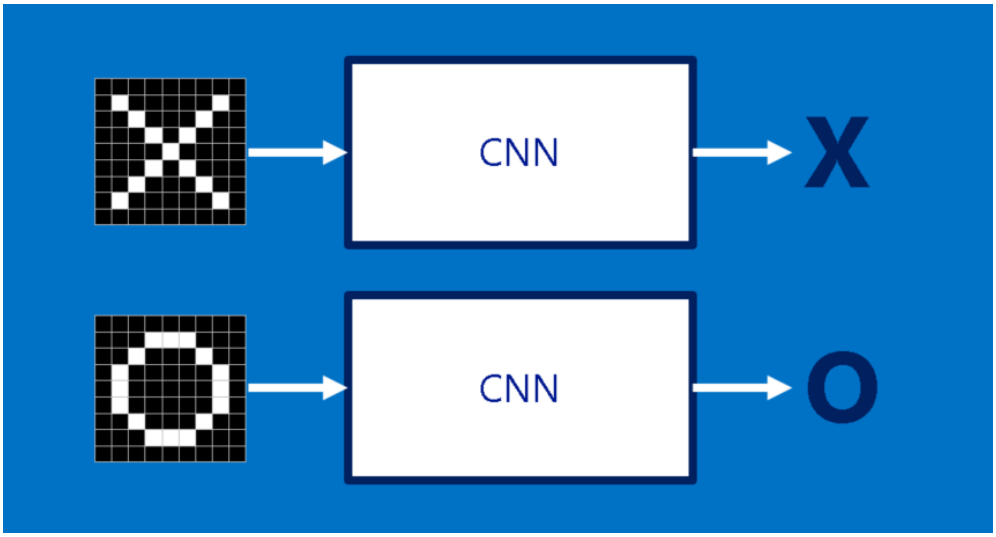
you can use this formula [(W−K+2P)/S]+1.

- W is the input volume - in your case 128
- K is the Kernel size - in your case 5
- P is the padding - in your case 0 i believe
- S is the stride - which you have not provided.

So, we input into the formula:

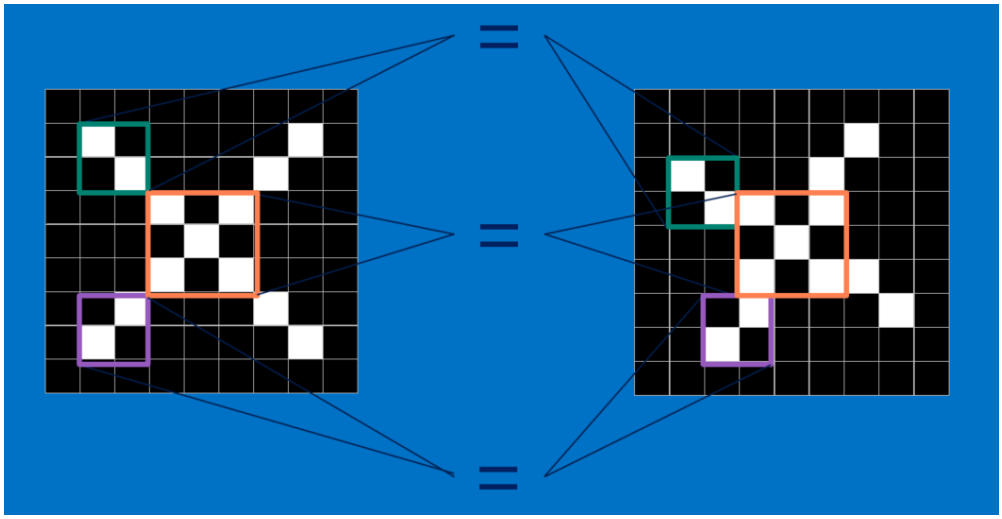
Output_Shape = (128-5+0)/1+1

Output_Shape = (124,124,40)

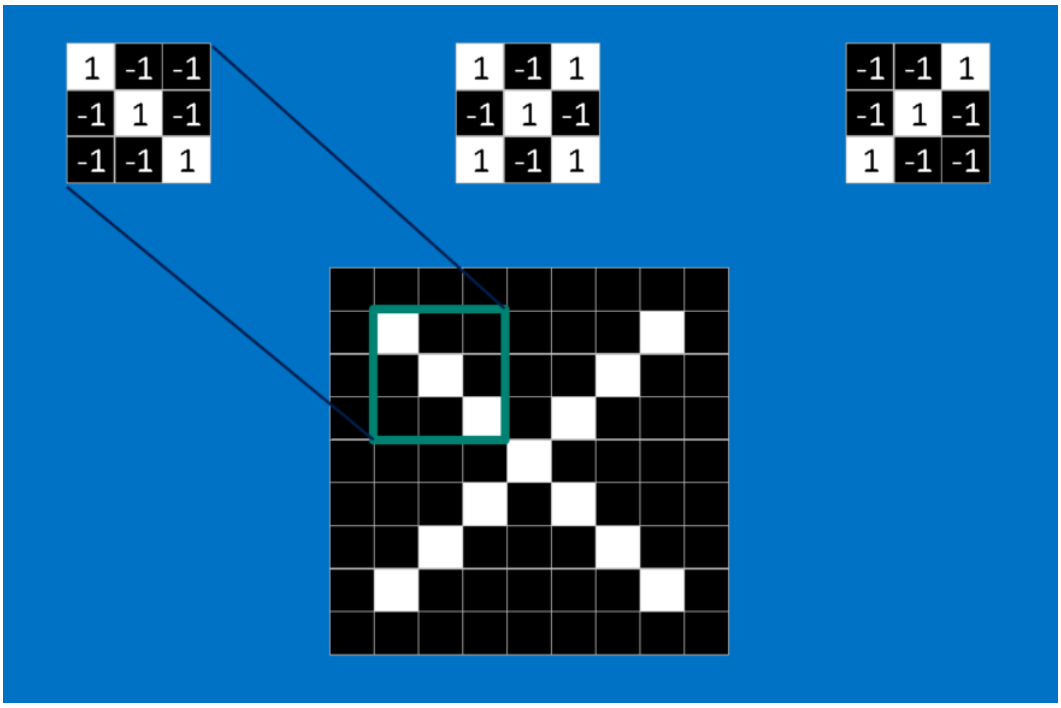


Để hiểu cách làm việc của CNNs, ta cùng quay lại ví dụ được nêu ra ở đầu bài viết: phân loại hình vẽ là X hay O

Feature

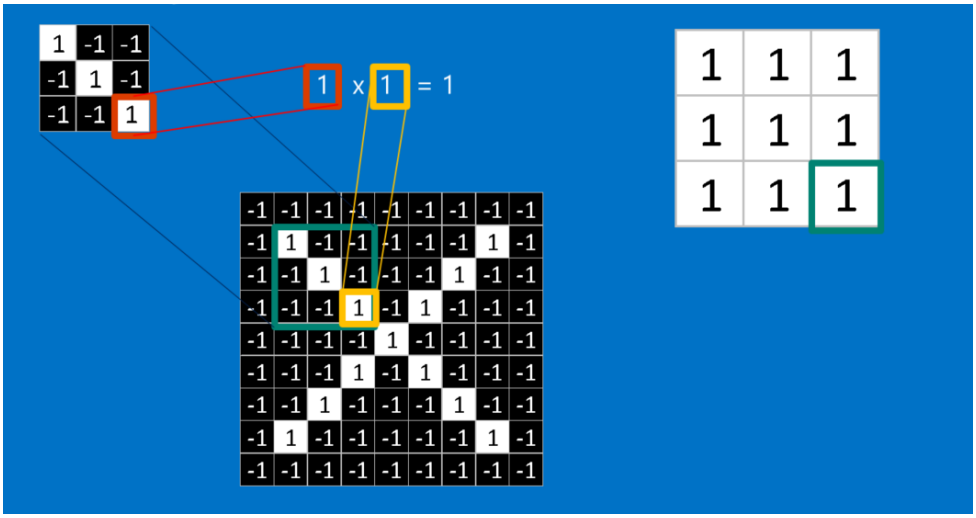


CNNs sẽ so sánh những tấm ảnh của chữ X theo từng mảnh. Những mảnh ảnh mà CNNs tìm được gọi là features. Bằng cách tìm những feature giống nhau ở vị trí tương ứng, CNNs đem đến kết quả tốt hơn khi so sánh với tất cả hình ảnh.

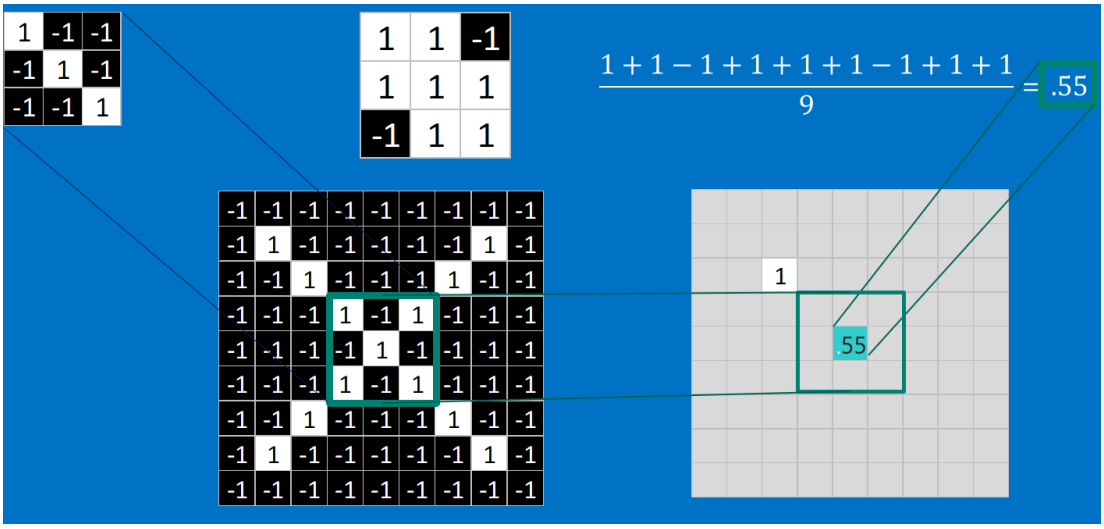


Mỗi feature giống như 1 bức ảnh mini và cũng là một mảng 2 chiều. Feature có đặc điểm là phản ánh được các khía cạnh của tấm ảnh. Ví dụ như trong bức ảnh chữ X, features sẽ bao gồm một mảnh ảnh đường chéo và một mảnh ảnh hình chữ thập. Hai đặc điểm đó giúp ta nhận ra đó chính là chữ X chứ không phải chữ O.

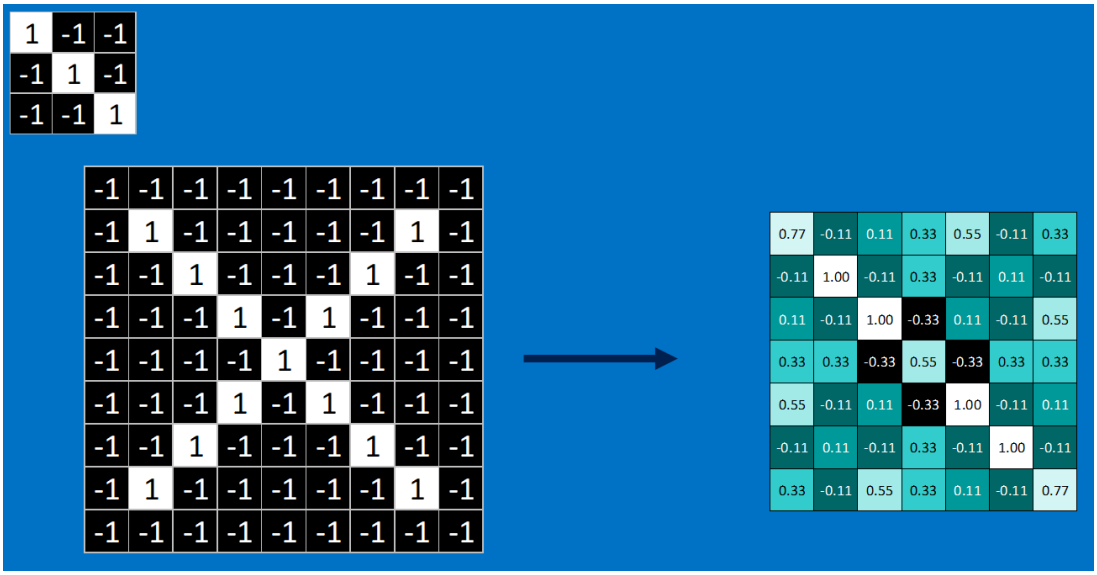
Convolutional



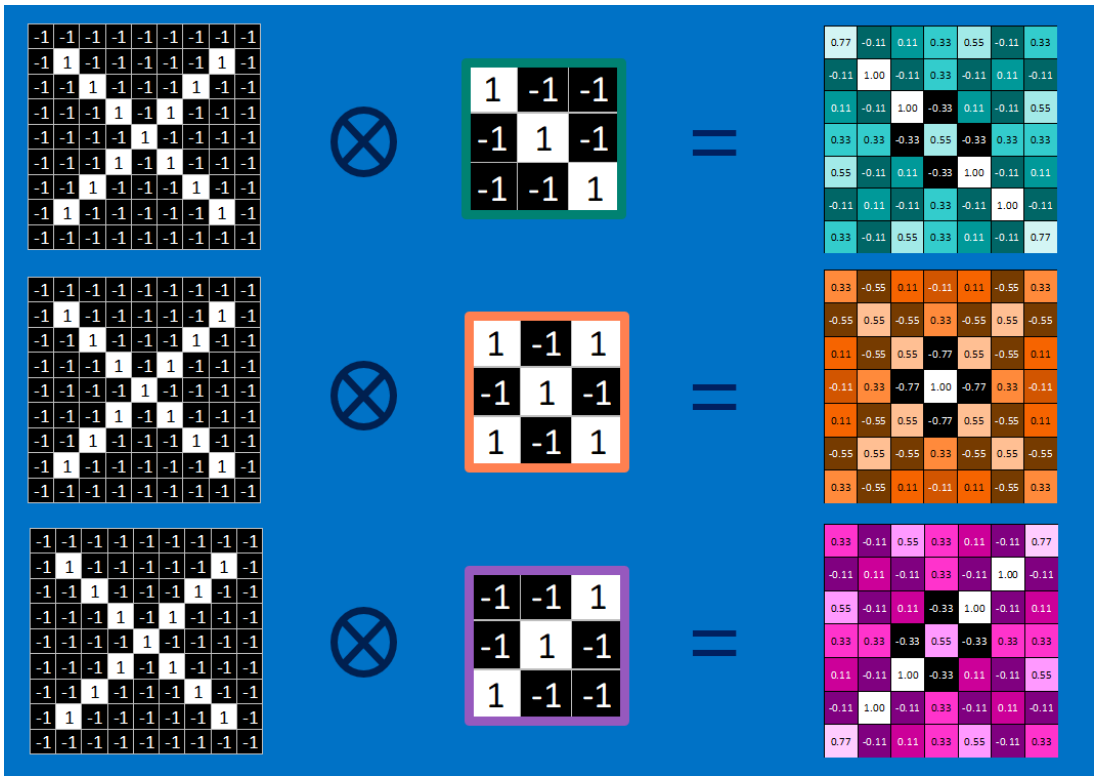
Khi nhìn vào một bức ảnh, CNNs sẽ không biết chính xác vị trí của những feature nên sẽ thử tất cả những vị trí khả thi. Và để tính toán các feature phù hợp cho toàn bộ các bức ảnh, ta gọi nó là một bộ lọc (filter).



Khi đưa bức ảnh qua bộ lọc, ta làm như sau: nhân vô hướng ma trận đường chéo 3x3 cho từng ma trận con trong bức ảnh, sau đó tính giá trị trung bình



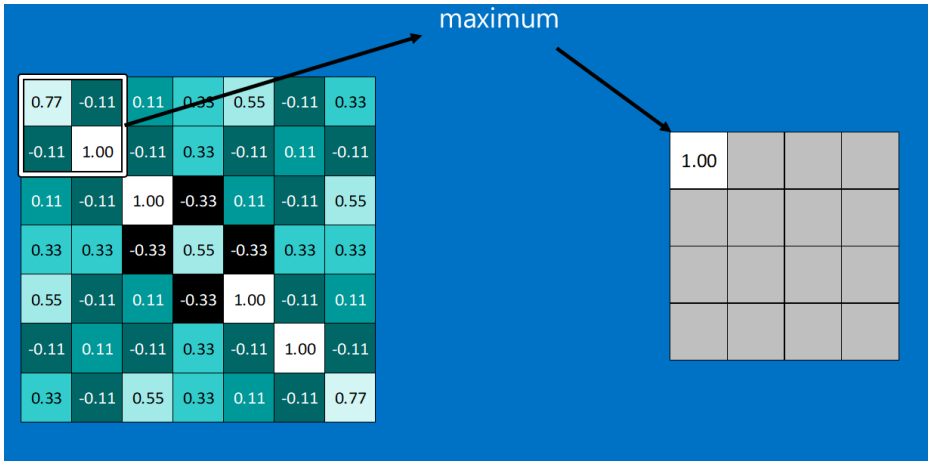
Làm điều tương tự với cả 3 bộ lọc



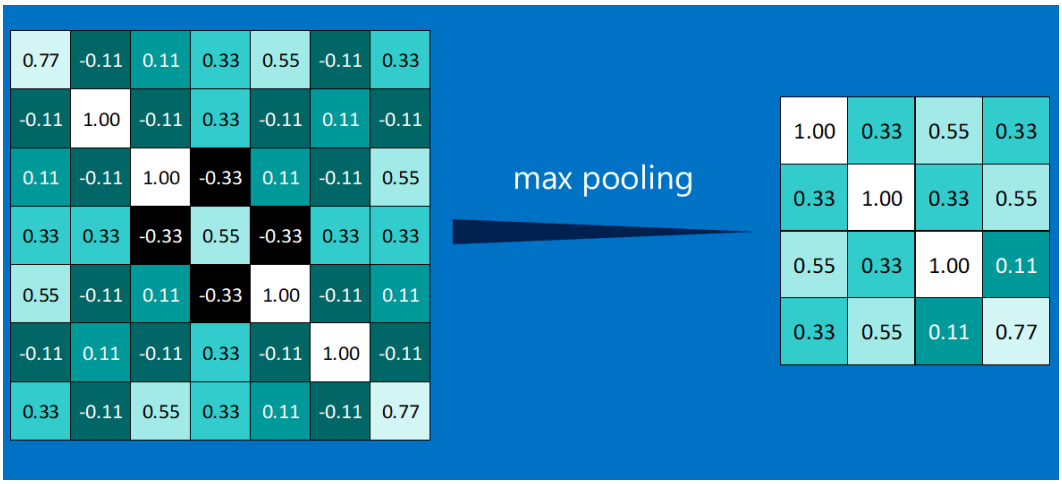
Và 3 filter màu tím, màu cam và màu hồng ở hình trên chính là ConvLayer

Pooling

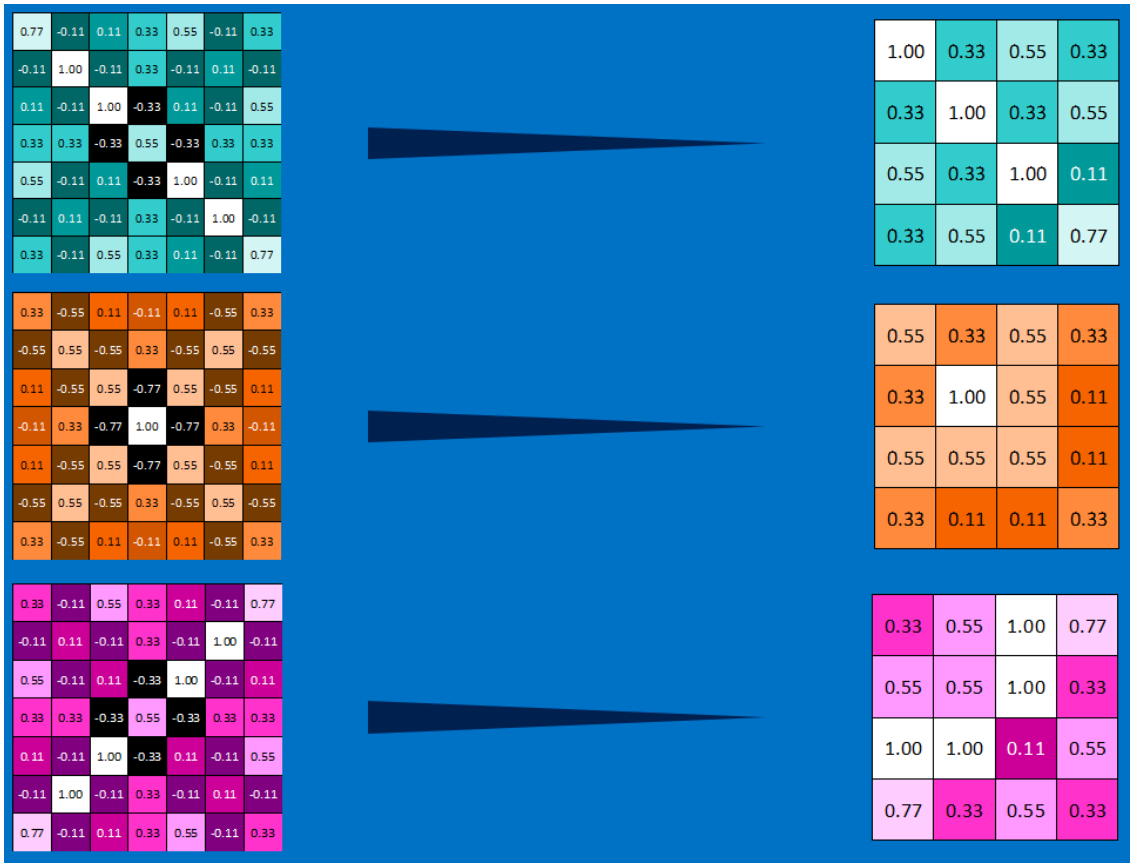
Sau khi đưa ảnh qua bộ lọc ta sẽ được output là các ma trận. Tiếp theo ta đưa những output đó qua lớp pooling, trong bài toán này ta sử dụng maxpooling, chọn window size = 2 và stride = 2



Ta thu được kết quả như sau

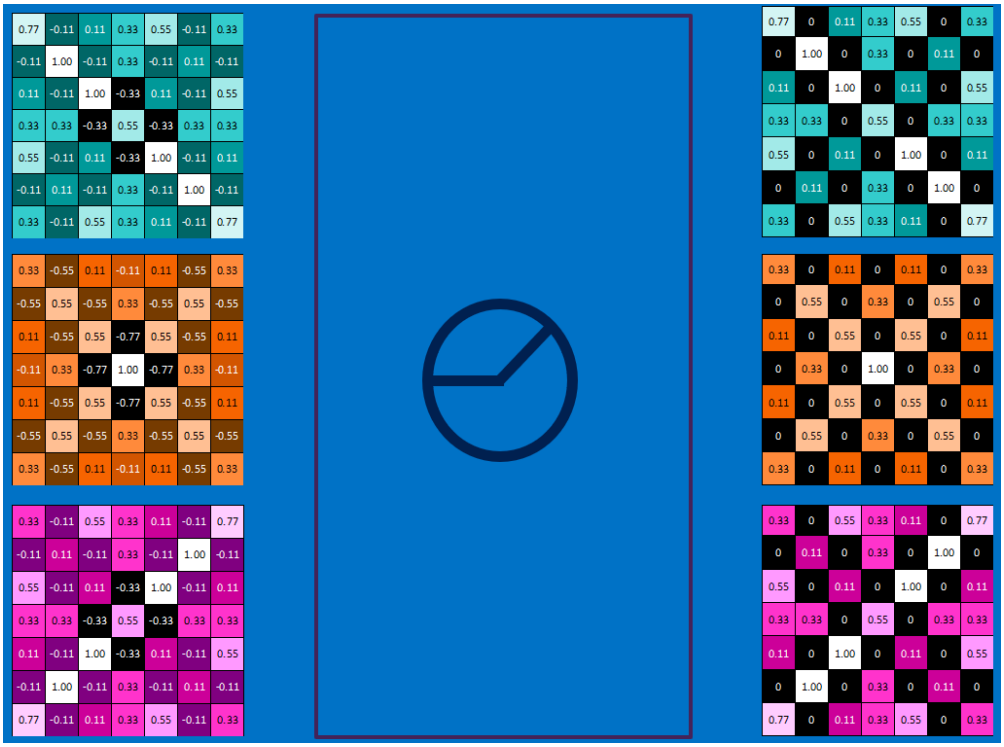


Làm điều tương tự với tất cả các bộ lọc

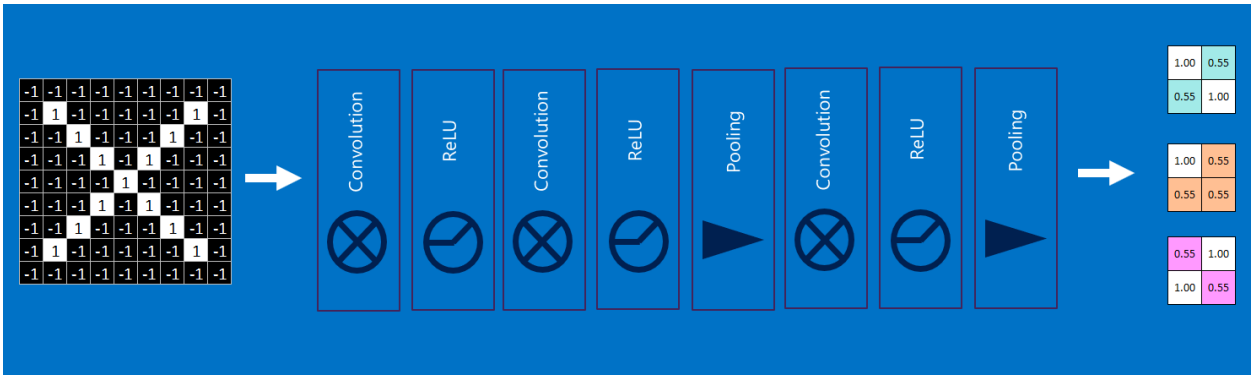


Rectified Linear Units

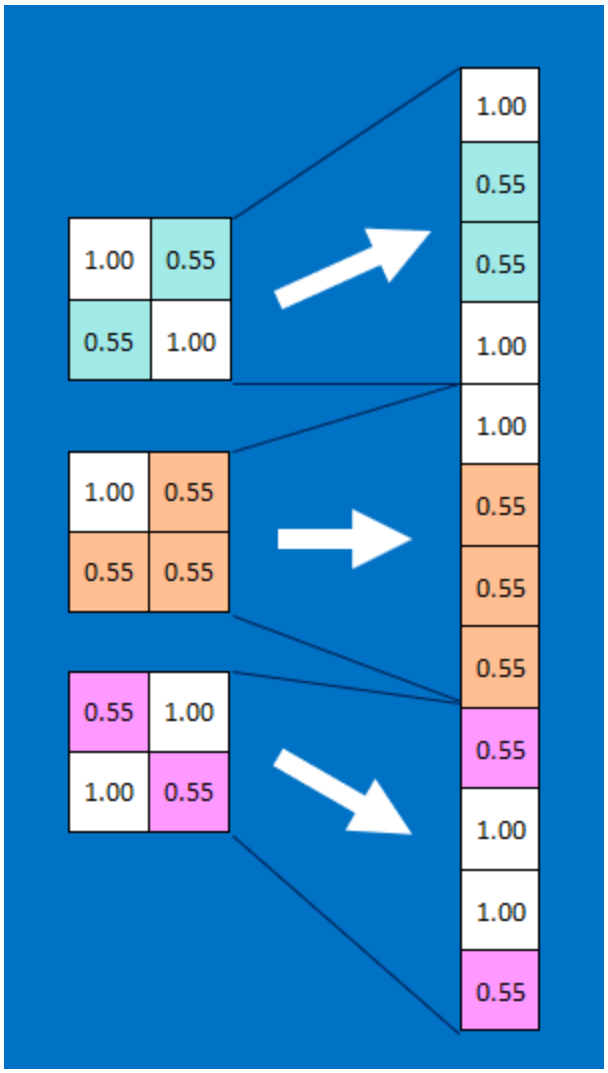
Tiếp theo, ta sẽ đưa những ma trận đã qua lớp pooling vào ReLu



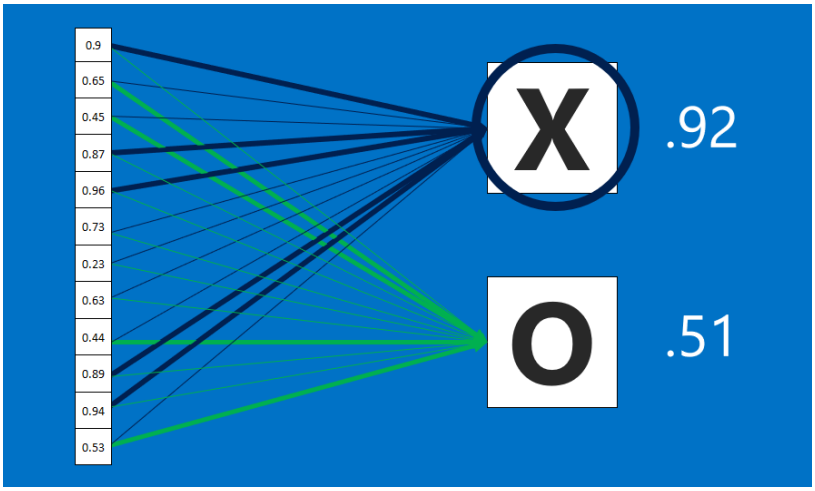
Nó lại tiếp tục thực hiện quá trình trên theo sơ đồ sau



Vậy là cuối cùng ta sẽ có 3 ma trận 2x2, việc tiếp theo là biến đổi 3 ma trận này thành 1 vector gồm 12 phần tử



Cuối cùng sẽ thực hiện việc tính toán dựa vào gradient descent và backprop



	Right answer	Actual answer	Error
X	1	0.92	0.08
O	0	0.51	0.49
		Total	0.57