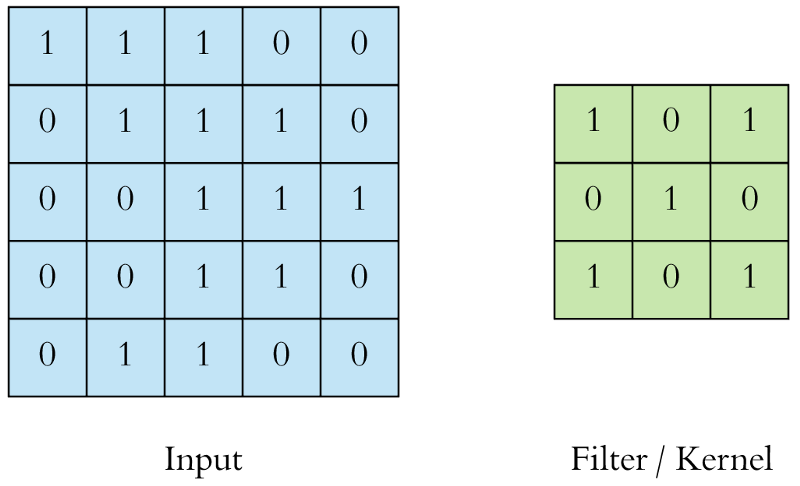
**Convolution**

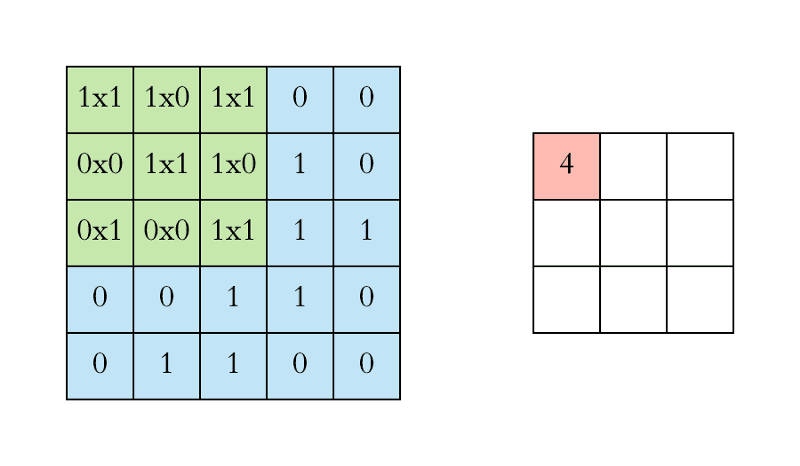
Gồm 2 khái niệm **Convolution Filter** và **Convolutional Layer**

Trong mạng neural network thông thường, từ input, ta cho qua các hidden layer rồi ra được output. Với CNN, **Convolutional Layer** cũng chính là hidden layer, khác ở chỗ, **Convolutional Layer** là một tập các **feature map** và mỗi **feature map** này là một bản scan của input ban đầu, nhưng được trích xuất ra các feature/đặc tính cụ thể. Scan như thế nào thì lại dựa vào **Convolution Filter** hay **kernel**. Đây là một ma trận sẽ 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 **kernel** rồi cộng tổng lại, đưa qua activation funciton (sigmoid, relu, elu, … ), kết quả sẽ là một con số cụ thể, tập hợp các con số này lại là 1 ma trận nữa, chính là **feature map**.

Ví dụ:

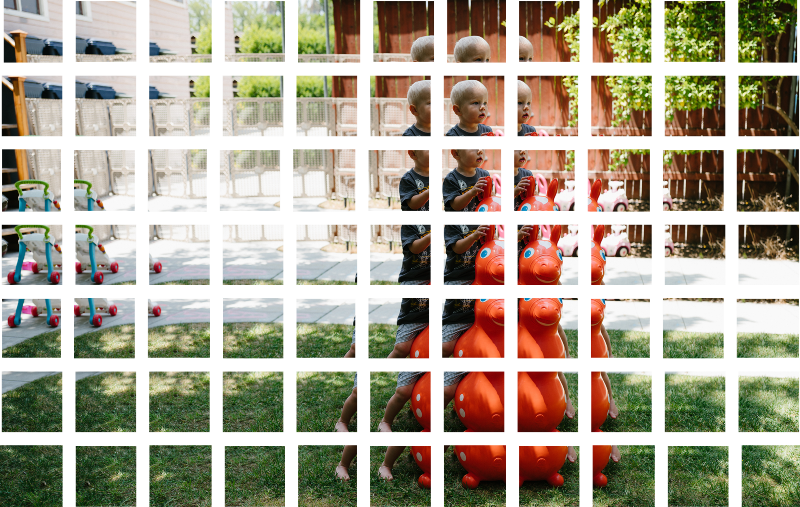


Quét **kernel** qua từng phần tử của input. Và tính toán như trên: nhân tương ứng, rồi cộng tổng kết quả, đưa qua activation function (ta bỏ qua bước này trong ảnh động biểu diễn dưới đây), ta thu được một giá trị tại **feature map**

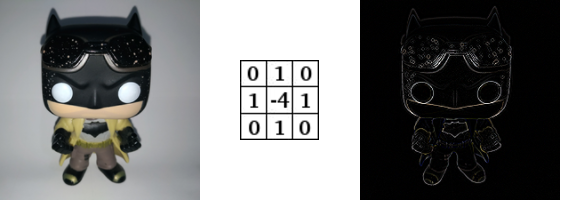


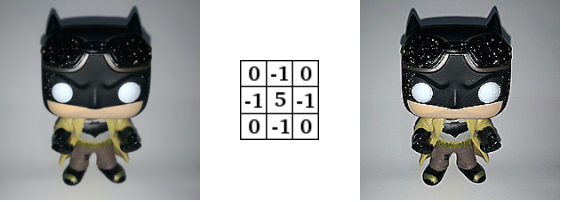
### Stride and Padding

**Stride** là khoảng cách giữa 2 **kernel** khi quét. **Padding là** ô bọc xung quanh các cạnh của input



Với mỗi **kennel** ta thu được **future** khác nhau tương ứng





CNN có nhiệm vụ tìm **kennel** và lưu **kennel**, ta không cần quan tâm đến.

CNN so sánh ảnh theo từng **future**, bằng cách tìm ở mức thô các **future** khớp với nhau ở cùng vị trí trong 2 bức ảnh, giải quyết bài toán phân loại ảnh ở mức chính xác cao

### Pooling

Mục đích của **pooling** rất đơn giản, nó làm giảm số hyperparameter mà ta cần phải tính toán, từ đó giảm thời gian tính toán, tránh overfitting. Pooling hoạt động gần giống với convolution, nó cũng có 1 cửa sổ trượt gọi là pooling window, cửa sổ này trượt qua từng giá trị của ma trận dữ liệu đầu vào (thường là các feature map trong convolutional layer), chọn ra **một** giá trị từ các gía trị nằm trong cửa sổ trượt (với max pooling ta sẽ lấy giá trị lớn nhất).

Sau khi pooling, một hình ảnh sẽ có khoảng một phần tư số điểm ảnh so với lúc bắt đầu. Vì nó giữ các giá trị lớn nhất từ mỗi cửa sổ, nó sẽ bảo toàn tính khớp của mỗi feature bên trong cửa sổ. Nghĩa là nó không quan tâm quá nhiều về vị trí chính xác nơi feature khớp, miễn là nó khớp ở chỗ nào đó trong cửa sổ. Kết quả là CNNs có thể tìm xem liệu một feature có nằm trong hình ảnh mà không cần lo nó nằm ở đâu. Điều này giúp giải quyết vấn đề của máy tính là quá trực nghĩa. Đầu ra sẽ có cùng số lượng hình ảnh, nhưng mỗi cái sẽ có điểm ảnh ít hơn. Điều này cũng rất hữu ích trong việc quản lý tải trọng tính toán. Hạ một tấm ảnh 8 megapixel xuống còn 2 megapixel sẽ giúp mọi xử lý tải về trở nên dễ dàng.

## **Fully connected layers**

Fully connected layers lấy các hình ảnh đã lọc ở convolution layer và pooling layer và chuyển chúng thành các phiếu bầu (vote). Sau đó một cuộc bầu chọn được tổ chứ, câu trả lời có nhiều phiếu nhất sẽ thắng và tuyên bố thuộc loại, đối tượng nào

## **Cấu trúc CNN**

