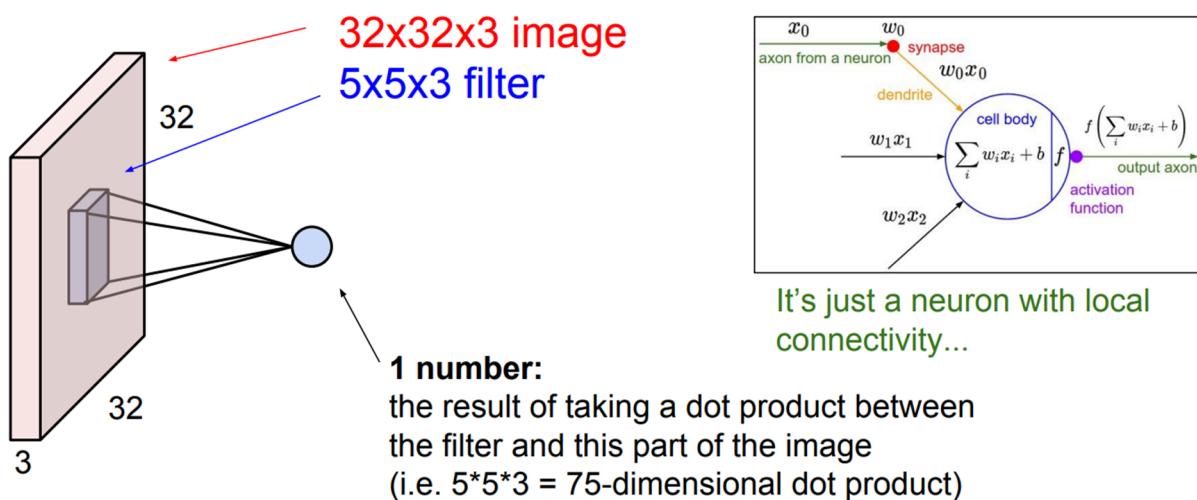


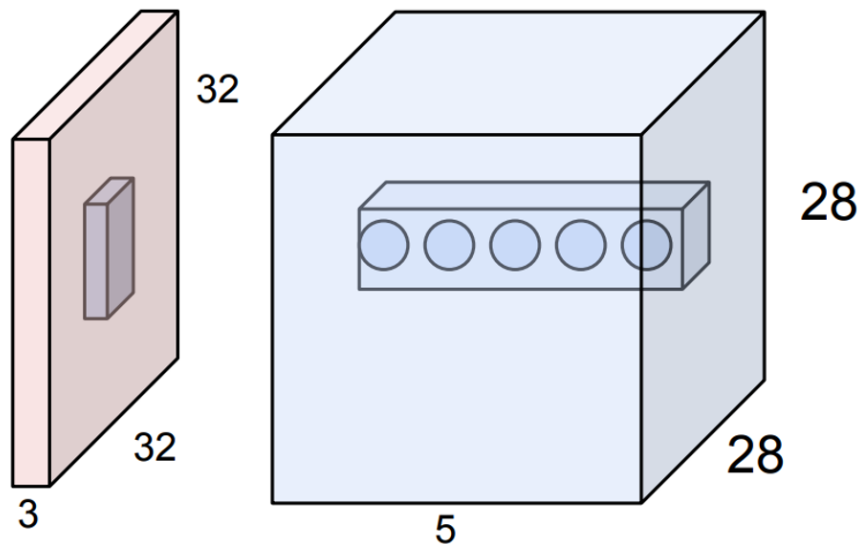
Lecture 5-4

Title	Convolutional Neural Networks
slide	http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture5.pdf

brain/neuron view



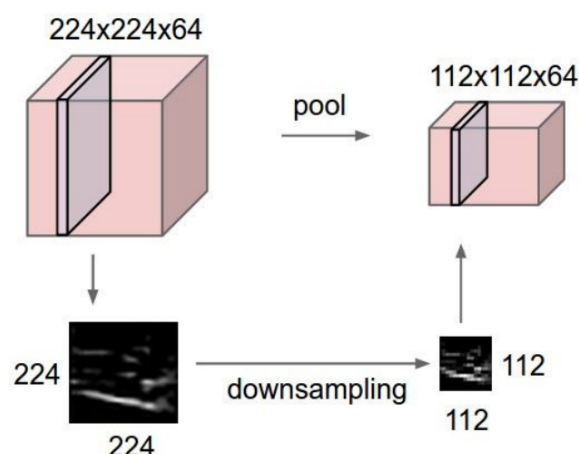
Conv Layer를 보면, 전체 이미지의 특정 위치에 필터를 가지고 내적을 수행하여 하나의 값을 얻게 된다. 이는 오른쪽 그림의 내적과 같은 아이디어로 표현할 수 있다. 입력이 들어오면 필터 값인 W s와 곱하고 하나의 값을 출력한다. 가장 큰 차이점은 우리의 뉴런은 Local connectivity를 가지고 있다는 것이다. Conv Layer처럼 슬라이딩을 하는게 아니라, 특정 부분에만 연결되어 있어서 하나의 뉴런은 한 부분만 처리하고, 그런 뉴런들이 모여서 전체 이미지를 처리한다. 이러한 방식으로 spatial structure를 유지한 채로 Layer의 출력인 activation map을 만드는 것이 가장 큰 차이점이다.



"Receptive field"란 한 뉴런이 한 번에 수용할 수 있는 영역을 의미한다. 가령 5x5 필터가 있다고 하면, 이는 곧 한 뉴런의 "Receptive field"가 5x5라고 한다. 만약 5개의 필터가 있다고 생각해보면, 출력 값은 위의 파란색 Volume처럼 28 x 28 x (필터의 개수)로 생기게 되므로, 28 x 28 x 5인 3D Grid가 된다. 그리고 나서 어떤 한 점을 찍어서 depth방향으로 바라보면 (파란색 Map 안에 5개 점), 각 필터는 서로 다른 특징을 추출하므로, 이 5개의 점은 정확하게 같은 지역에서 추출된 서로 다른 특징이라고 할 수 있다.

지난 시간에 공부한 FullyConnected Layer를 다시 생각해보면, FC-Layer는 32 x 32 x 3을 모두 편 다음 전체를 연결하여 계산한다. 반면에 Conv Layer는 전체가 아직 지역 정보만 이용한다는 점에서 차이가 있다.

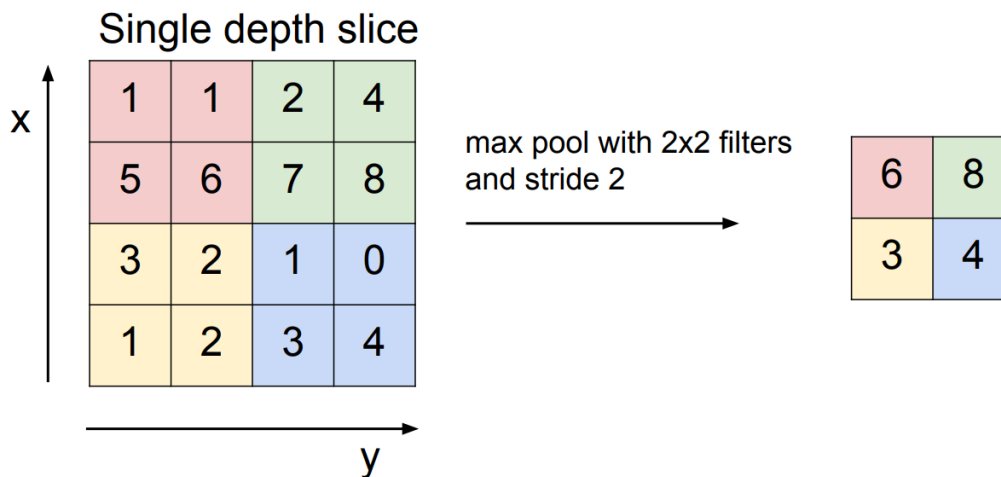
Pooling layer



Pooling도 일종의 Stride 기법으로, 요새는 Downsample할 때 pooling을 하기보단 stride를 많이 사용하고 있는 추세이고, 실제로 요즘은 stride가 더 좋은 성능을 보이고 있다. Pooling Layer는 Representation들을 더 작게 만들고, representation이 작아지면 파라

미터의 수가 줄게 되어 관리하기도 더 쉬워진다. 또한 Pooling Layer은 Downsample 하여 일종의 공간적인 불변성(invariance)을 얻을 수 있다. 예를 들어 $224 \times 224 \times 64$ 인 입력이 있다면, 이를 $112 \times 112 \times 64$ 로 "공간적"으로 줄여준다. 여기서 중요한 점은 "Depth"에는 아무 짓도 하지 않기 때문에 Depth에는 영향을 주지 않는다는 점이다.

Max Pooling



Pooling은 다양한 종류가 있는데 일반적으로 Max Pooling을 많이 사용한다. 우리가 다루는 값들은 얼마나 뉴런이 활성화 되었는지. 즉, 이 필터가 각 위치에서 얼마나 활성화 되었는지이다. Max pooling은 그 지역이 어디든, 어떤 신호에 대해 "얼마나"그 필터가 활성화 되었는지를 알려준다. 가령 그 값이 어디에 있었다는 것 보다, 그 값이 얼마나 큰 지가 중요 것과 같다.

예를 들어, 위와 같이 2x2 필터가 있고 Stride=2라고 하면, Conv Layer가 했던 것처럼 슬라이딩하면서 연산을 수행한다. 대신에 여기서는 내적을 하는 것이 아니라, 필터 안에 가장 큰 값 중에 하나를 고른다. 빨간색 영역을 보면 6이 제일 크고, 초록색 영역은 8이 제일 크기 때문에 오른쪽과 같이 표현할 수 있다.