

# Lecture 5-5

Title	Convolutional Neural Networks
slide	<a href="http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture5.pdf">http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture5.pdf</a>

## Design Choice

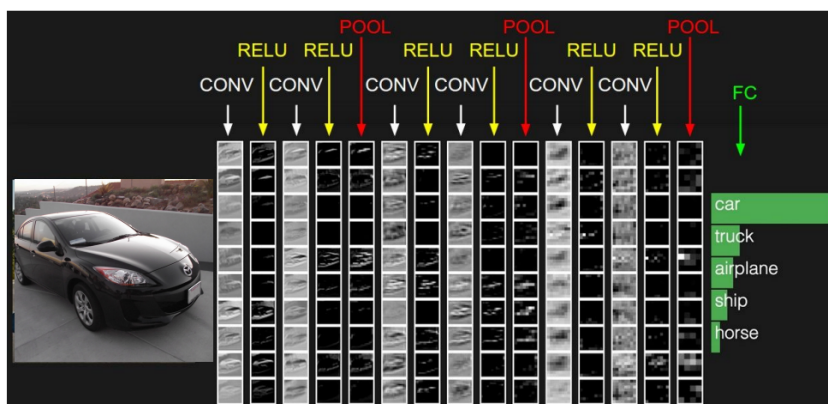
Common settings:

- Accepts a volume of size  $W_1 \times H_1 \times D_1$
- Requires three hyperparameters:
  - their spatial extent  $F$ ,
  - the stride  $S$ ,
- Produces a volume of size  $W_2 \times H_2 \times D_2$  where:
  - $W_2 = (W_1 - F)/S + 1$
  - $H_2 = (H_1 - F)/S + 1$
  - $D_2 = D_1$
- Introduces zero parameters since it computes a fixed function of the input
- Note that it is not common to use zero-padding for Pooling layers

$F = 2, S = 2$

$F = 3, S = 2$

Pooling Layer에는 몇 가지 Design Choice가 있다. 입력이 W(width), H(Height), D(Depth)라면 이를 통해 Filter Size를 정해줄 수 있다. 그리고 여기에 Stride까지 정해주면, 앞서 Conv Layer에서 사용했던 수식을 그대로 이용해서  $(W - \text{Filter Size}) / \text{Stride} + 1$ 로 Design Choice를 할 수 있다. pooling layer의 한 가지 특징은 보통 padding을 하지 않는다는 것이다. 우리는 downsampling하고 싶고, Conv와 같이 코너의 값을 계산하지 못하는 경우도 없기 때문에 Pooling 할 때는 padding을 고려하지 않고 그냥 downsample만 하면 된다. 가장 널리 쓰이는 필터 사이즈는 2x2, 3x3이고 stride는 2이다. 필터가 3x3 일 때도 일반적으로 stride는 2x2 로 한다.



위의 이미지에서 각 열들의 위에는 POOL같은 것들이 써있는데, 각 요소들은 출력 Activation map이다. 정리해보자면, Conv Layer와 ReLU Layer거치고, downsampling을

하고 싶을 때는 Pooling을 섞어주고 마지막으로 FC Layer를 거친다. 여기서 FC Layer은 이전에 살펴본 FC와 같다. 마지막 Conv Layer의 출력은 3차원 volume으로 이루어지는데, 이 값들을 모두 펴서(stretch) 1차원 벡터로 만들고 FC Layer의 입력으로 사용한다. 그렇게 되면 Conv Net의 모든 출력을 하나로 통합시키고 최종적인 추론을 하여 Score가 출력한다. 따라서 이 마지막 Layer부터는 공간적 구조(spatial structure)를 신경 쓰지 않게 되게 된다.

오른쪽 맨 끝에 있는 Pool Layer의 출력 값은 전체 네트워크를 통과한 집약체라고 할 수 있다. 그러므로 우리가 만든 계층 구조의 최상위라고 할 수 있고, 실제로 Higher level concept을 표현하고 있는 것들이다. 최하위 계층에서는 edges와 같은 단순한 구조를 찾아내므로, 여기 첫번째 열(col)의 map들이 의미하는 것은 각 자리에서 edges같은 것들이 얼마나 존재하는지를 의미한다. 그리고 지나면 지날수록 더 복잡한 것들을 찾아내어, 두 번째 레이어에서는 (edge보다 복잡한) corner같은 것이 얼마나 있는지를 보여준다. 마지막 pooling layer를 거치기 전까지의 각 값들은 각 필터가 가진 template이 얼마나 활성화되었는지를 표현하고, 이런 정보를 가지고 FC Layer를 거치게 되면, 그 정보들을 한데 모아 클래스 스코어를 계산하는 것이다.

## Summary

- ConvNets stack CONV, POOL, FC layers
- Trend towards smaller filters and deeper architectures
- Trend towards getting rid of POOL/FC layers (just CONV)
- Typical architectures look like  
 **$[(\text{CONV-RELU}) * N - \text{POOL}] * M - (\text{FC-RELU}) * K, \text{SOFTMAX}$**   
where N is usually up to ~5, M is large,  $0 \leq K \leq 2$ .
  - but recent advances such as ResNet/GoogLeNet challenge this paradigm

전형적인 CNN 아키텍처는 Conv와 ReLU를 몇 번(n번) 반복하고 그 중간에 pooling도 몇 번 해준 후, FC Layer가 한 번에서 두 번 정도 이어진다. 그리고 Class score를 구하기 위해 softmax를 사용한다. 이 때, 네트워크의 필터는 점점 더 작아지고, 아키텍처는 점점 더 깊어지는 경향이 있다는 것을 배웠다. 요즘에는 Pooling이나 FC Layer를 없애고, 그냥 Conv Layer만 깊게 쌓는 것이 추세이다.