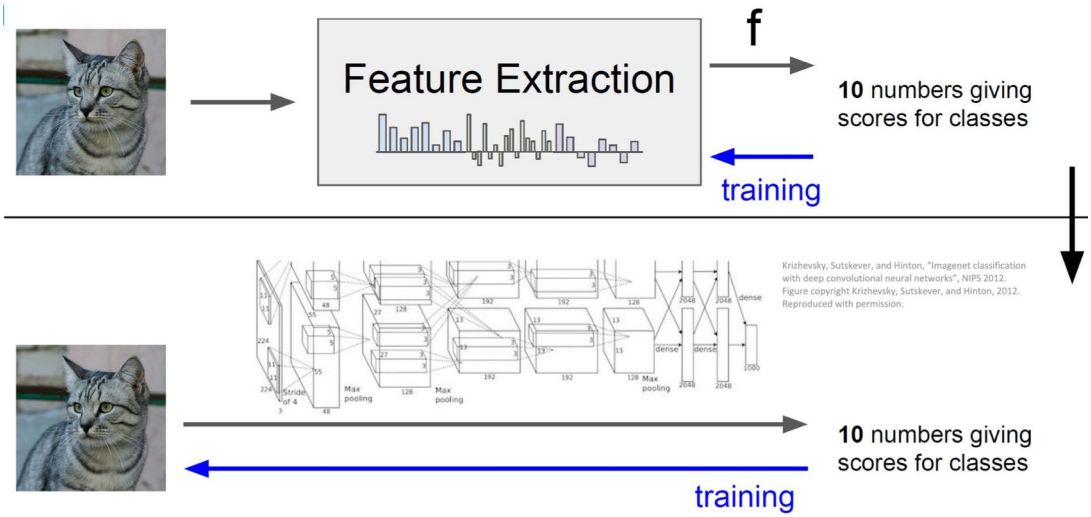
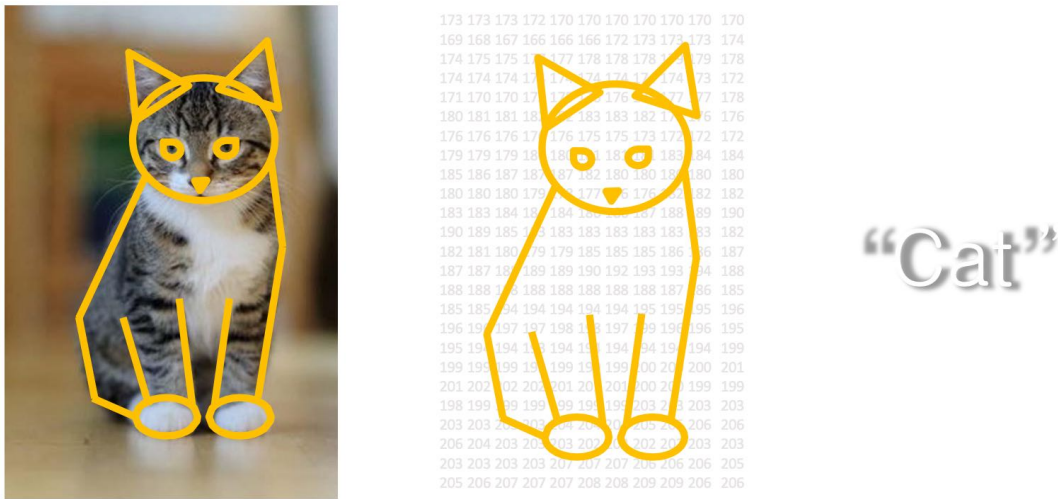
CNN

Machine Learning vs Deep Learning

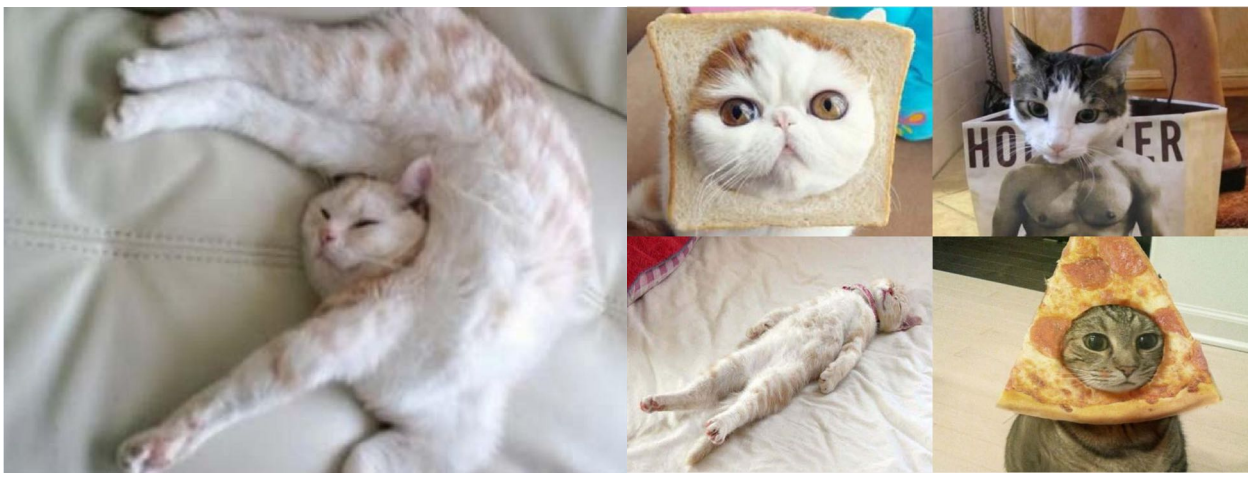


이미지로부터 Filter 개수 등의 Hyperparameter를 조절하여 Feature Extraction을 자동화

ANN (Image Analysis)



Problem



In image analysis, each pixel is highly correlated to pixels surrounding it (i.e. neighbors)

Fully Connected ? topology 변화에 대응이 어려움

- **모든 weight 계산**(이미지의 **모든 픽셀의 연관관계를 한번에 모두 연산) > 비효율성**
- **위치(가로 x 세로) 개념? > 위치에 민감**
  똑같은 pixel은 다른 위치에 있더라도 똑같은 feature로 input되어야하는데, **위치가 바뀐 동일 feature(pixel)에 대해서 hidden-Layer에서 동일한 feature을 중복해서 뽑아냄**

Solution

locally-connected

- **특정 픽셀에 대해서 거리가 가까운 pixel은 연관성이 있어서 계산**
- **거리가 먼 pixel은 연관성이 없는 것으로 보고서 계산하지 않는 방식**

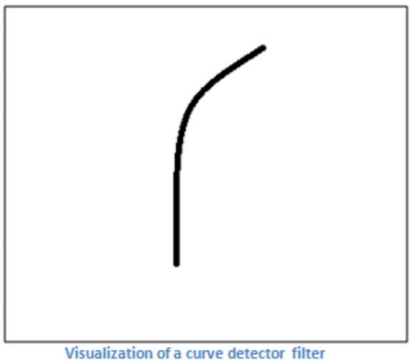
Convolutional Filter

이미지가 **locally-conntected한 이미지 데이터라는 것을 가정**해야 나타나는 개념
(현재위치의 1pixel과 그 **가로세로의 크기만큼의 주변pixel의 연관관계를 계산 - 가로x세로의 Kernel=Filter를 sliding 하면서 연산)**

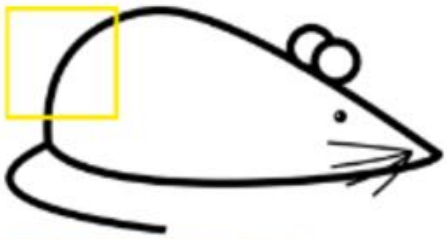
input의 위치와 관계없이, shared weight(특정 영역 추출)를 sliding window하고, 그것을 연산(dot product)하여, 똑같은 feature라면 위치와 상관없이 해당feature를 뽑아낼 수 있음

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter



Visualization of a curve detector filter



Visualization of the filter on the image

Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Filter

Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)

Visualization of the filter on the image

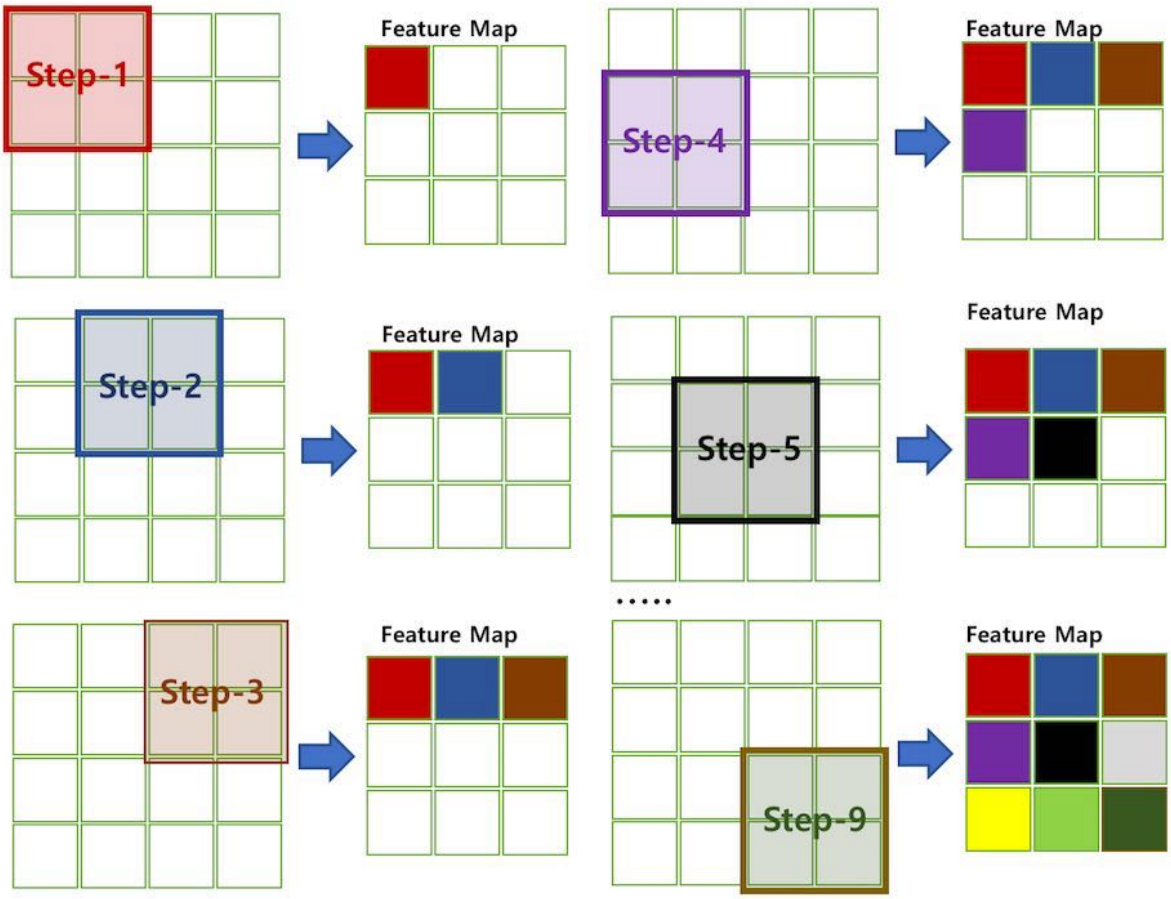
0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

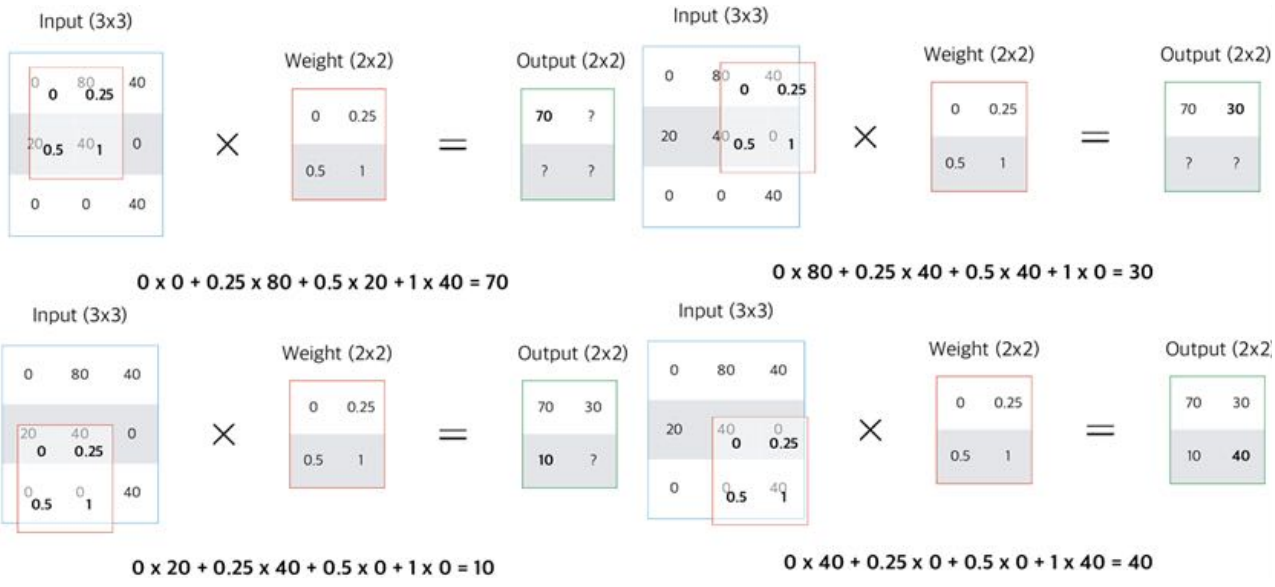
Pixel representation of receptive field

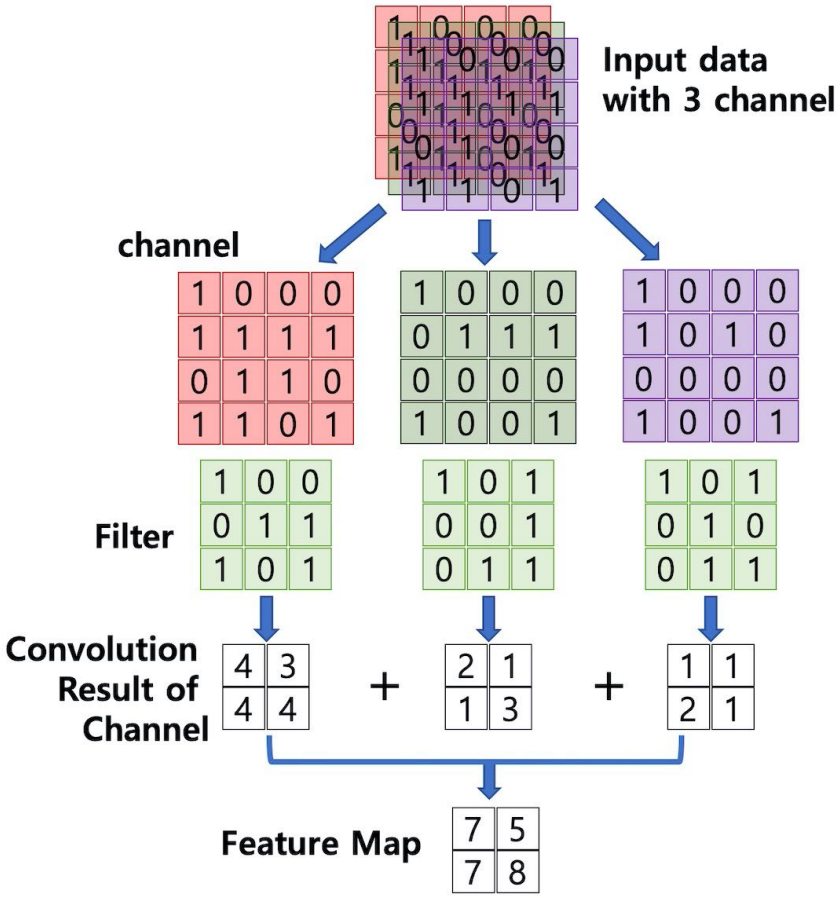
*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter


Step-1
Feature Map
Step-4
Feature Map
Step-2
Feature Map
Step-5
Feature Map
Step-3
Feature Map
Step-9
Feature Map


Input (3x3)
Weight (2x2)
Output (2x2)
0 x 0 + 0.25 x 80 + 0.5 x 20 + 1 x 40 = 70
Input (3x3)
Weight (2x2)
Output (2x2)
0 x 80 + 0.25 x 40 + 0.5 x 40 + 1 x 0 = 30
Input (3x3)
Weight (2x2)
Output (2x2)
0 x 20 + 0.25 x 40 + 0.5 x 0 + 1 x 0 = 10
Input (3x3)
Weight (2x2)
Output (2x2)
0 x 40 + 0.25 x 0 + 0.5 x 0 + 1 x 40 = 40

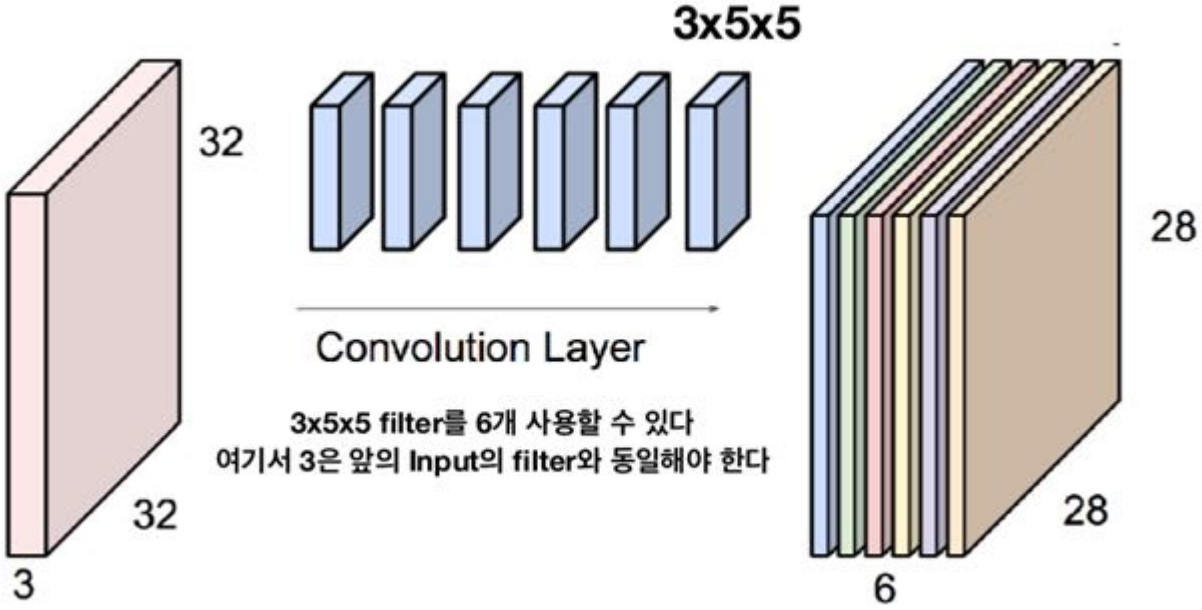


- Feature Map은 합성곱 계산으로 만들어진 행렬
- Activation Map은 Feature Map 행렬에 활성 함수를 적용한 결과

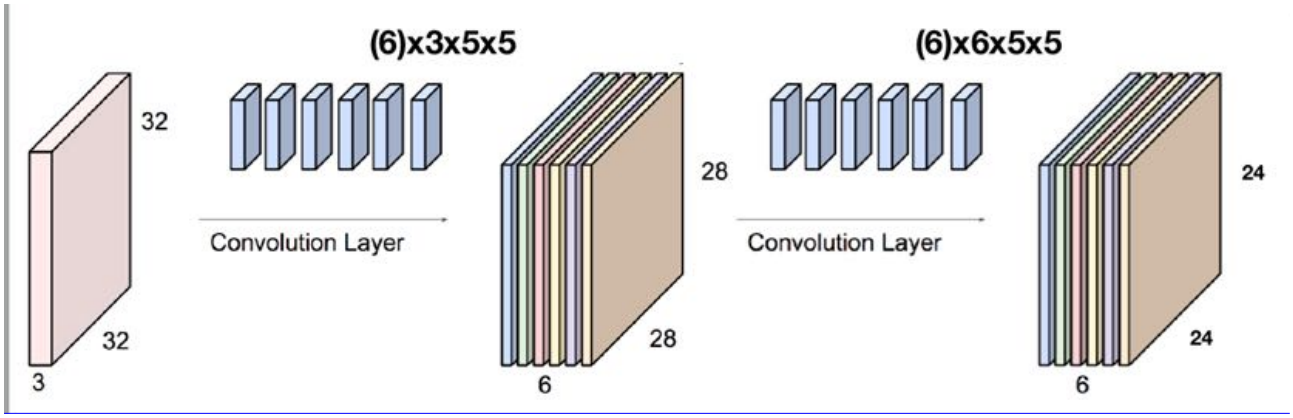
즉 Convolution 레이어의 최종 출력 결과가 Activation Map

Convolutional Layer

하나의 Filter당 하나의 feature를 뽑아냄. 그러므로 여러 개의 feature를 뽑아내기 위해서는 여러개의 Filter를 써야한다. 여러개의 Convolution Filter를 묶어놓은 것을 Convolution Layer라고 함

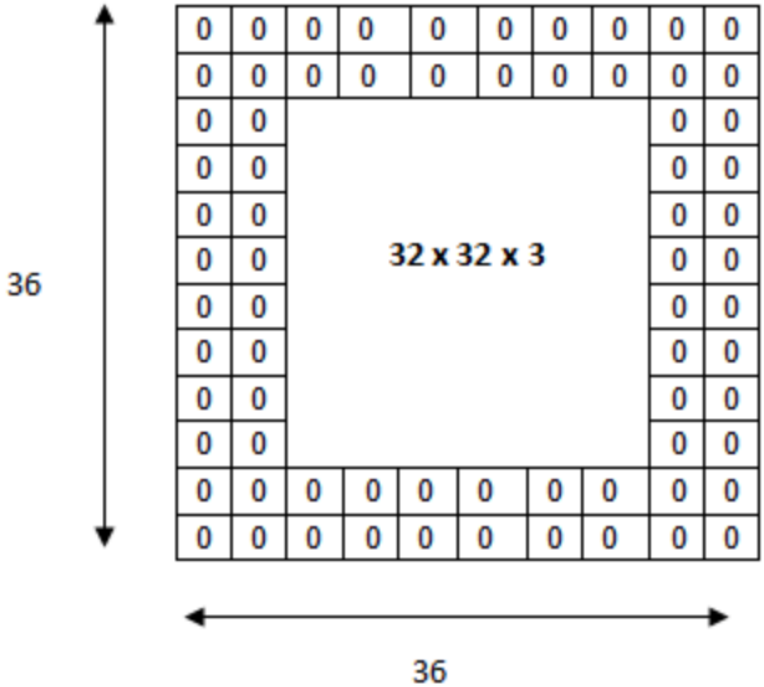


기본적으로 weight처럼 **6개의 Convolutional Filter는 서로 다른 랜덤 초기화 필터**
각 **Convolutional Filter는 6개가 각각 연산**해서 각각 **6개의 output을 만들고** 그 **output은 새로운 feature로서** 6개가 쌓이는데 **output feature**라고 부른다.



- input_filter_size와 convolutional filter의 filter_size는 같아야 함
- output_filter_size와 convolutional Layer의 filter개수는 같아야 함

Padding = Sub sampling



Padding을 통해서 Convolution 레이어의 출력 데이터의 사이즈를 조절
외각을 "0"값으로 둘러싸는 특징으로 부터 인공 신경망이 이미지의 외각을 인식
Padding은 결과 값을 작아지는 것을 막아서 특징이 유실되는 것을 막는 것 뿐 아니라, 오버피팅도 방지하게 되는데, 원본 데이타에 0 값을 넣어서 원래의 특징을 희석 시켜 버리고, 이것을 기반으로 머신러닝 모델이 트레이닝 값에만 정확하게 맞아 들어가는 오버피팅 현상을 방지

Pooling

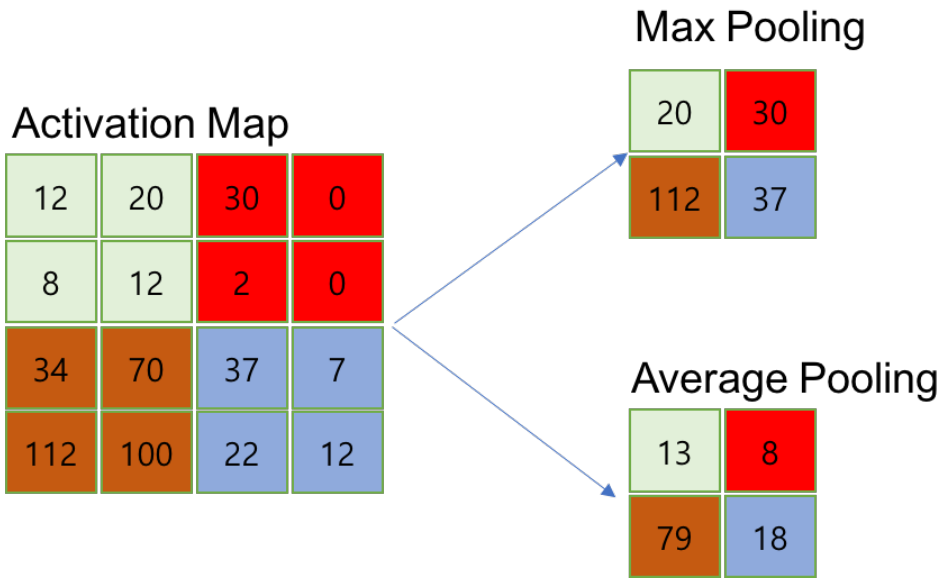
데이터의 사이즈를 강제로 줄이는 개념. Max Pooling은 Max한 큰값만 챙기고 작은 값을 버림

특징을 나타내는 픽셀은 그렇게 많기 때문에 중요한 pixel만 선별해서 연산

즉, 필요없는 feature를 버려서, 남은 메모리는 다음 Conv Layer를 쌓는데 쓰는, 연산을 효율적으로 만듦

전체 데이타의 사이즈가 줄어들기 때문에 연산에 들어가는 컴퓨팅 리소스가 적어지고

데이타의 크기를 줄이면서 소실이 발생하기 때문에, 오버피팅을 방지

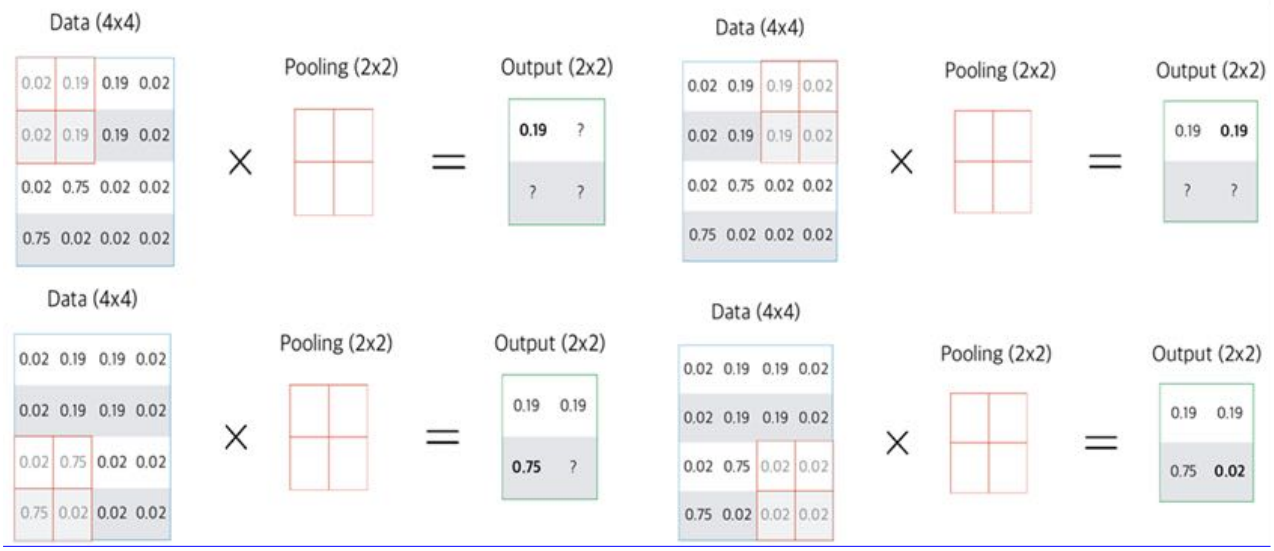


Pooling 레이어는 Convolution 레이어와 비교

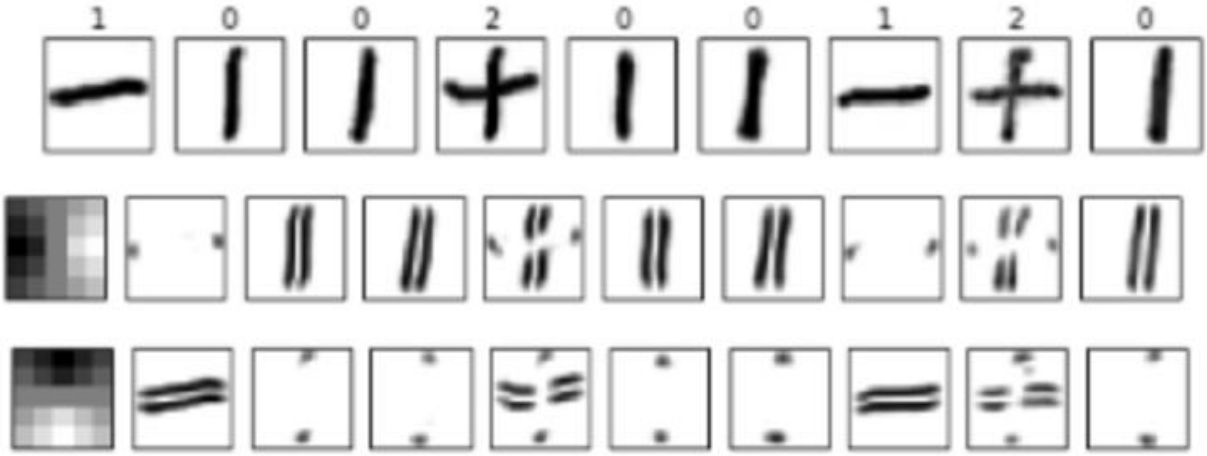
- 학습대상 파라미터가 없음
- Pooling 레이어를 통과하면 행렬의 크기 감소
- Pooling 레이어를 통해서 채널 수 변경 없음

Max Pooling

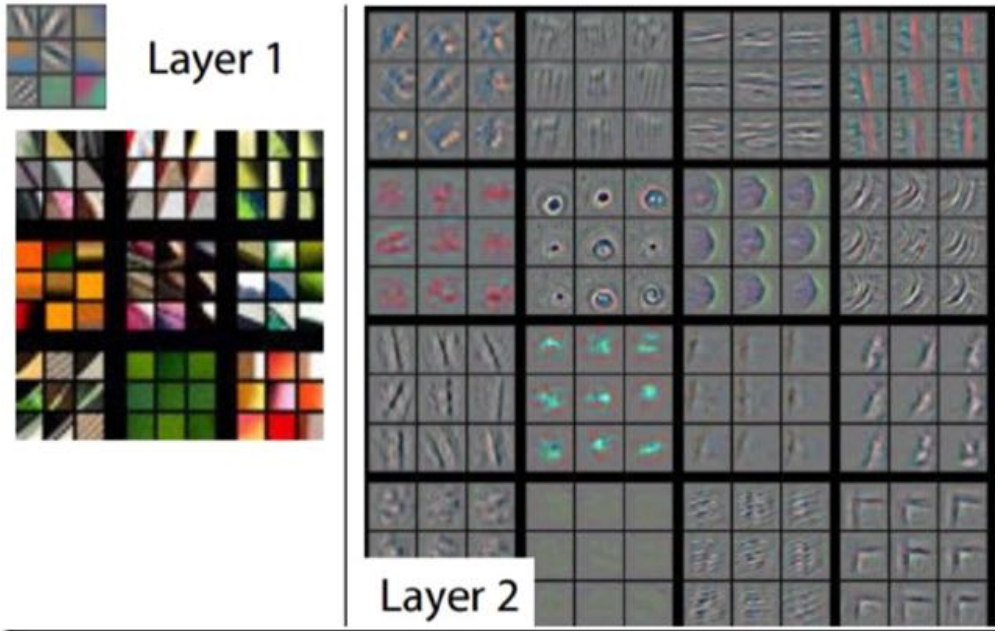
실용적인 접근에 있어서 많이 안쓰려고 하는 추세



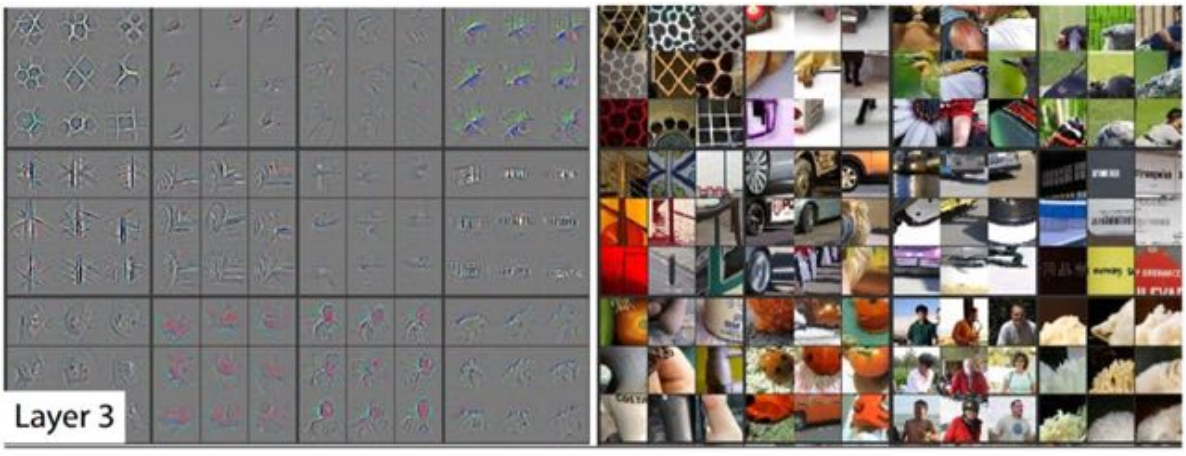
Convolution 이해



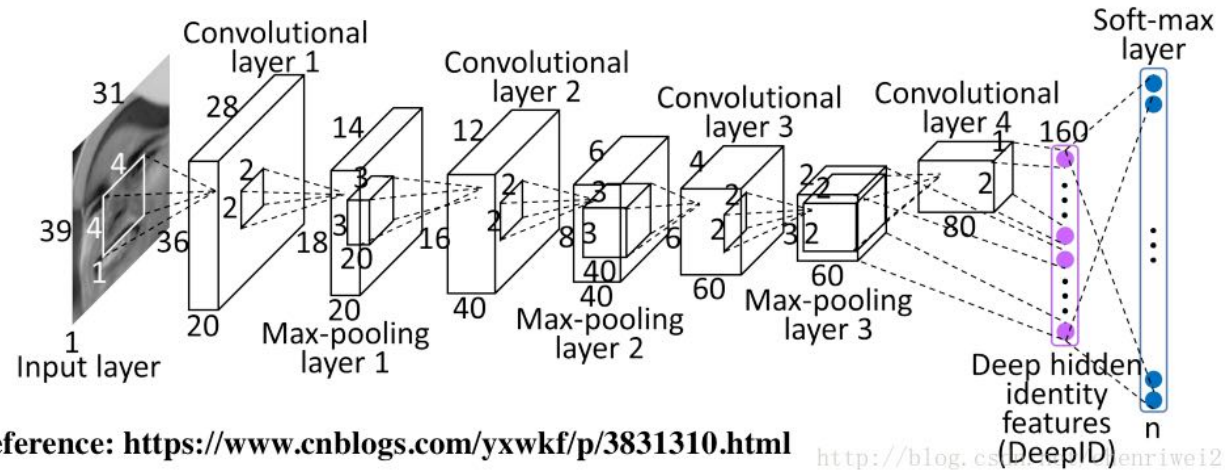
- C.N.N.의 Layer는 앞쪽일수록 저차원의 linear한 feature를 뽑아내고, 뒤로 갈수록 고차원의 feature를 뽑아냄
  - Layer1은 선형을, Layer2는 타원모양으로 점점 feature를 뽑아내는 것을 알 수 있음



Layer3에서는 벌집모양의 feature를 뽑아낸다. **마지막 Layer에서는 자동차class에서 자동차의 특징인 본넷과 타이어를** 뽑아내는 것을 알 수 있음



Parameter



Reference: https://www.cnblogs.com/yxwkf/p/3831310.html

layer	input channel	Filter	output channel	Stride	Pooling	활성함수	Input Shape	Output Shape	파라미터 수
Convolution Layer 1	1	(4, 4)	20	1	X	relu	(39, 31, 1)	(36, 28, 20)	320
Max Pooling Lyaer 1	20	X	20	2	(2, 2)	X	(36, 28, 20)	(18, 14, 20)	0
Convolution Layer 2	20	(3, 3)	40	1	X	relu	(18, 14, 20)	(16, 12, 40)	7,200
Max Pooling Lyaer 2	40	X	40	2	(2,2)	X	(16, 12, 40)	(8, 6, 40)	0
Convolution Layer 3	40	(2, 2)	60	1	1	relu	(8, 6, 40)	(6, 4, 60)	21,600
Max Pooling Lyaer 3	60	X	60	(2, 2)	60	X	(6, 4, 60)	(3, 2, 60)	0
Convolution Layer 4	60	(2, 2)	80	1	1	relu	(3, 2, 60)	(2, 1, 80)	19,200
Flatten	X	X	X	X	X	X	(2, 1, 80)	(160, 1)	0
fully connected Layer	X	X	X	X	X	softmax	(160, 1)	(100, 1)	160,000
합계	X	X	X	X	X	softmax	(160, 1)	(100, 1)	208,320

학습 파라미터 = "입력채널수X필터폭X필터높이X출력채널수입력채널수X필터폭X필터높이X출력채널수"

Summary

CNN(Convolutional Neural Network)은 이미지의 공간 정보를 유지하면서 인접 이미지와의 특징을 효과적으로 인식하고 강조하는 방식으로 이미지의 특징을 추출하는 부분과 이미지를 분류하는 부분으로 구성
특징 추출 영역은 Filter를 사용하여 공유 파라미터 수를 최소화하면서 이미지의 특징을 찾는 Convolution 레이어와 특징을 강화하고 모으는 Pooling 레이어로 구성
CNN은 Filter의 크기, Stride, Padding과 Pooling 크기로 출력 데이터 크기를 조절하고, 필터의 개수로 출력 데이터의 채널을 결정합니다.
CNN는 같은 레이어 크기의 Fully Connected Neural Network와 비교해 볼 때, 학습 파라미터양은 20% 규모입니다. 은닉층이 깊어질 수록 학습 파라미터의 차이는 더 벌어짐. CNN은 Fully Connected Neural Network와 비교하여 더 작은 학습 파라미터로 더 높은 인식률을 제공

- 각 레이어의 입출력 데이터의 형상 유지
- 이미지의 공간 정보를 유지하면서 인접 이미지와의 특징을 효과적으로 인식
- 복수의 필터로 이미지의 특징 추출 및 학습
- 추출한 이미지의 특징을 모으고 강화하는 Pooling 레이어
- 필터를 공유 파라미터로 사용하기 때문에, 일반 인공 신경망과 비교하여 학습 파라미터가 매우 적음