



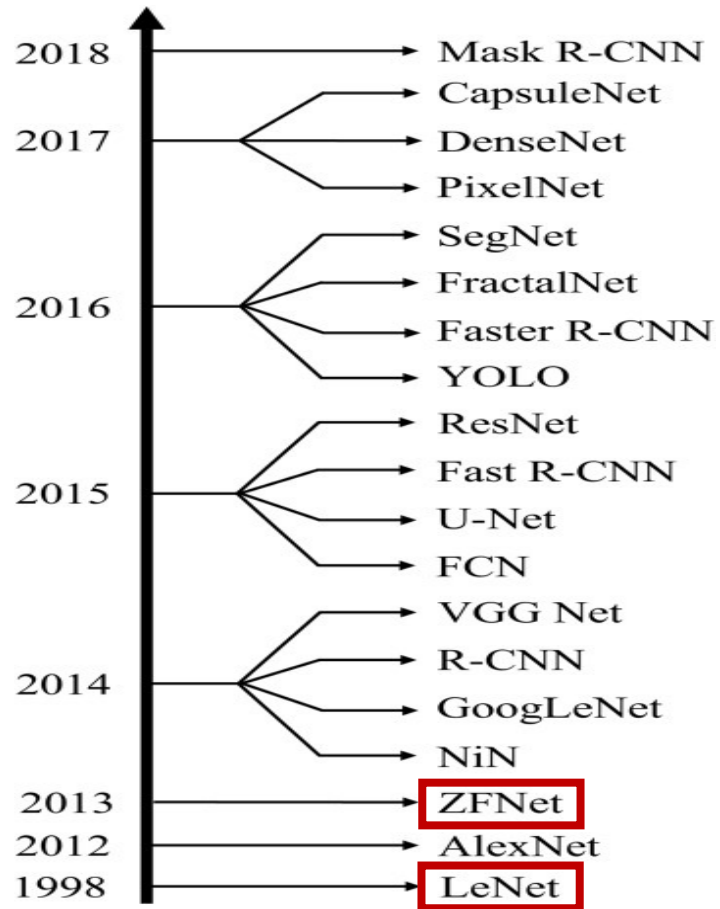
Convolutional Neural Network

LeNet & ZFNet

2022.02.14

황성아

Abstract



LeNet-5(1988)

- CNN의 초창기 모델
- Yann LuCun 박사가 손글씨 이미지를 학습하기 위해 만듦
- LeNet1,2,3,4,5 가 발표되었는데, 통상 5를 LeNet이라 함

ZFNet(2013)

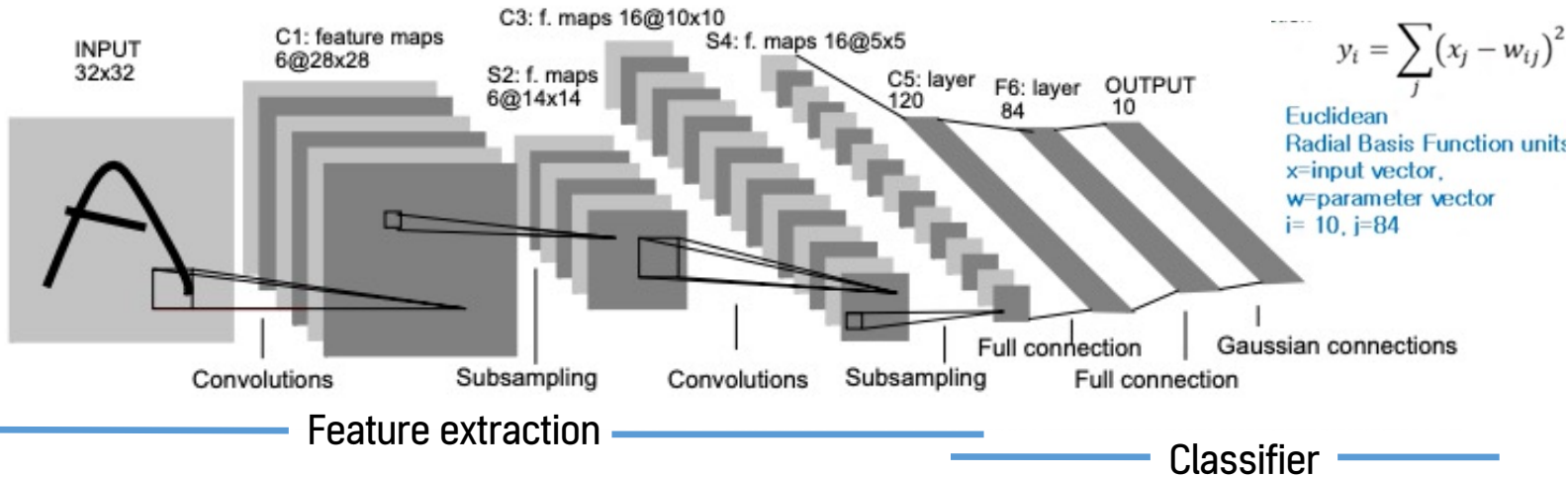
- "ILSVRC 2013"에서 우승
- 새로운 구조를 만든다는 것 보다 가시화 해 개선 방안을 파악하는 것이 목적
- Alexnet 에서 살짝 변화

LeNet 추가

[논문 리뷰] LeNet5 (1988) [Lecture]

Gradient-Based Learning Applied to Document Recognition, Yann LeCun [Paper]

LeNet-5 Architecture



H : 입력 데이터의 Height
P : Padding 사이즈
FH : Filter(Kernal)의 Height
FW : Filter(Kernal)의 Width
S : Stride 크기

Convolution Layer 출력 데이터 크기 계산 공식

OutputHeight : $[(H + 2P - FH) / S] + 1$

OutputWidth : $[(W + 2P - FW) / S] + 1$

Pooling Layer 출력 데이터 크기 계산 공식

OutputRowSize = InputRowSize / PoolingSize

OutputColumnSize = InputColumnSize / PoolingSize

* 현재 진행하는 모든 작업의 Padding의 사이즈는 0, stride는 1

Input의 크기가 32*32일때

C1을 얻기 위해 5*5 필터를 적용하면 $\rightarrow (32 + 0 - 5 + 1) = 28 * 28$

S2는 PoolingSize가 2*2 이기 때문에 14*14가 되고 커널 수는 유지

C3을 얻기 위해 다시 5*5 필터를 적용하면 $\rightarrow (14 + 0 - 5 + 1) = 10 * 10$

S4는 PoolingSize가 2*2 이기 때문에 5*5로 나올 수 있고 커널 수는 유지

C5에서는 FC(Full Connection)이 적용

우선 Convolution작업을 한다.

$(5 + 0 - 5 + 1) = 1 * 1$ 이고 커널 수는 120

이후 FC작업을 통해 일렬로 배치를 하고 84, 10으로 줄여감

RBF

• $\phi = \phi(r)$

- $r = \|x - x_i\|$ - Euclidian distance
- x_i - centers
- Gaussian RBF - $\phi = e^{-\varepsilon^2 r^2}$
- ε_i - shape parameters

• RBF Interpolant, given by a linear combination :

$$\phi(x) = \sum_{i=1}^N \lambda_i \phi(\|x - x_i\|, \varepsilon_i)$$

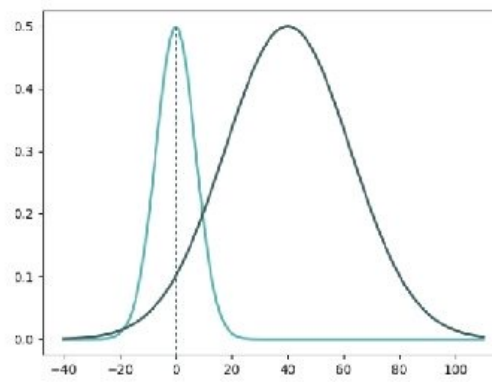


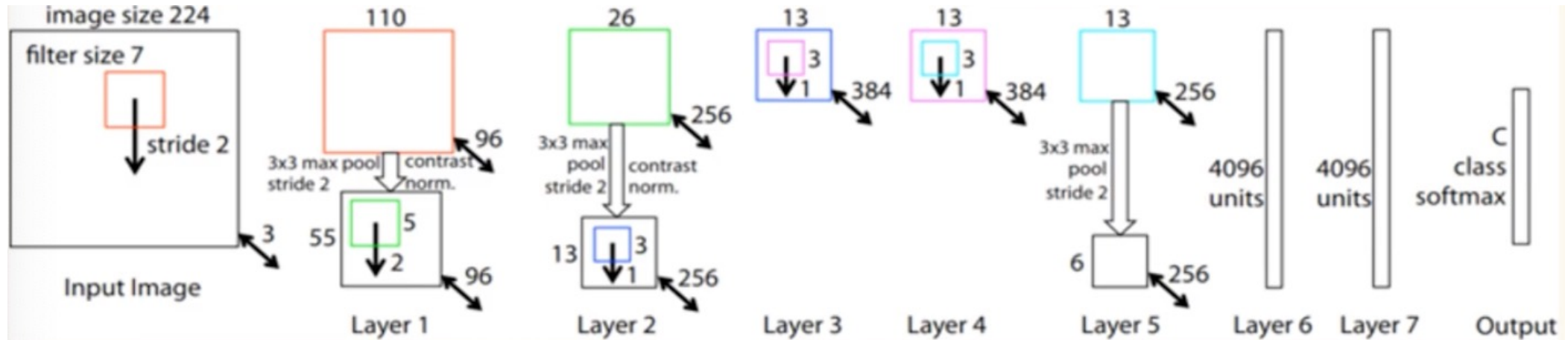
Figure 1.
 ■ $\varepsilon^2=0.01$
 ■ $\varepsilon^2=0.001$

- 방사형 기저 함수 (RBF)는 중심으로부터의 거리에 의해서만 정의되는 함수입니다. 정확한 위치는 중요하지 않습니다. 상대적 위치 만 중요합니다.
- 주로 RBF는 한 가지 속성 때문에 사용됩니다. 중앙에서 출력 (영향력)이 가장 높습니다. 중심에서 떨어진 각 거리 단위에서 (어느 방향 으로든) 영향이 감소합니다.
- RBF 커널은 각 지점의 중심에 방사형 기저 함수를 배치 한 다음 선형 조작을 수행하여 지점을 분리하기 더 쉬운 고차원 공간에 매핑합니다.
- Radial Basis Networks는 RBF 뉴런 레이어 1 개와 출력 뉴런 레이어 1 개가있는 단순한 2 계층 아키텍처입니다. RBF 뉴런에는 입력 벡터가 비교되는 '중앙 벡터'가 각각 할당됩니다. 이러한 네트워크는 기본적으로 밀도를 활용하므로 매우 작은 구조로 복잡한 비선형 성을 모델링 할 수 있습니다.

ZFNet

[Paper Review] Introduction to Image Classification Basic Networks [Lecture]

ZFNet Architecture



- AlexNet 에서 GPU를 2개-> 1개로 변경
- 일부 convolutional layer의 kernel size 와 stride 조절

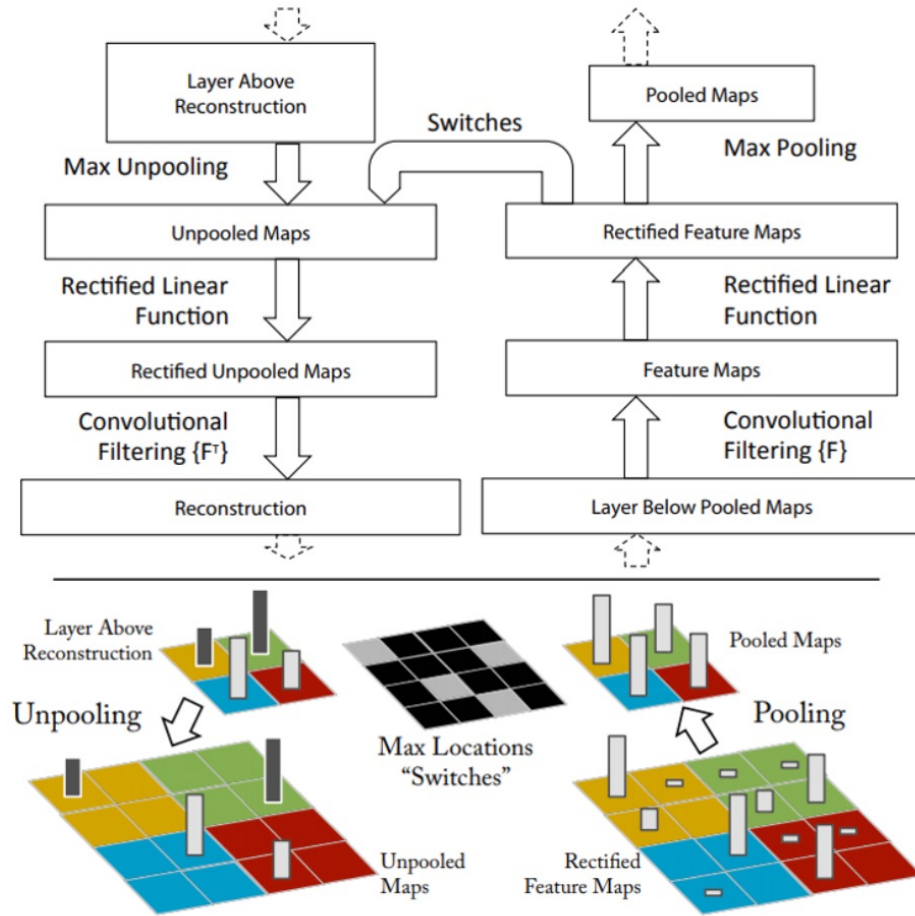
AlexNet:

2개의 GPU를 이용한 병렬 구조 학습 + **[11 by 11]filter에 stride 4**, [5 by 5], [3 by 3], [3 by 3] 순으로 convolution filter 적용

ZFNet:

1개의 GPU를 이용한 학습 + **[7 by 7]filter에 stride 2**, [5 by 5], [3 by 3], [3 by 3] 순으로 convolution filter 적용

Visualizing Method



Cnn은 convolution 연산 후 활성화 함수를 통해 feature map 을 생성하고 pooling 을 하여 이미지 축소 시키는 것 반복

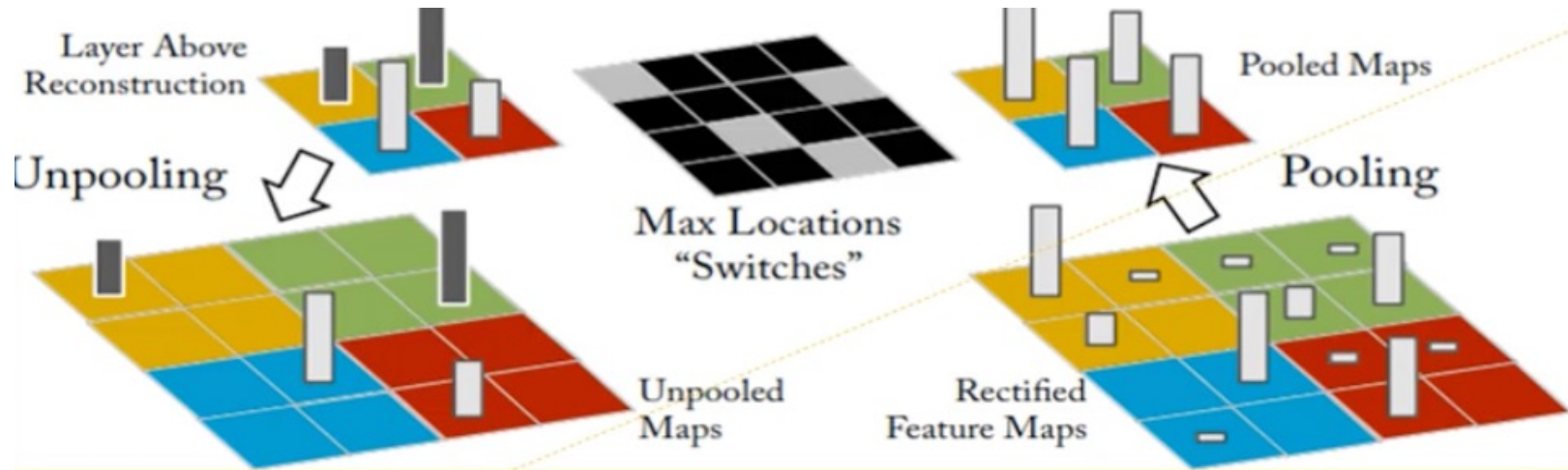
Zfnet은 이런 특정 feature들이 입력 이미지에 어떻게 mapping이 되는지 이해하기 위해 **역으로 수행**하는 과정 거침

Maxpooling 대신 unpooling
Activation 대신 rectification
Convolution 대신 deconvolution

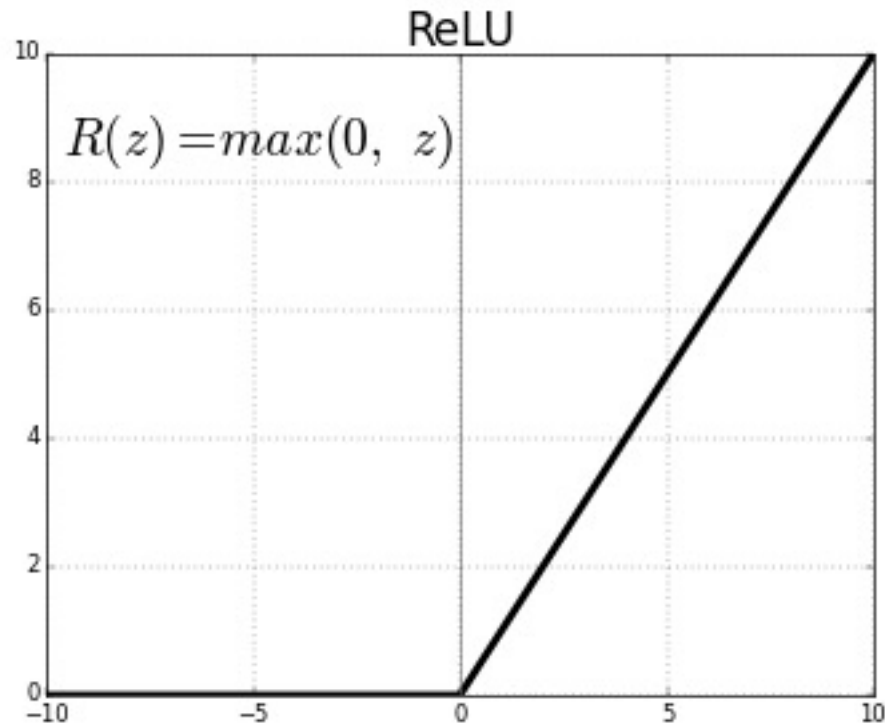
Visualizing Method: Unpooling

Maxpooling 단계에서는 주변에서 강한 자극만 다음으로 전달하기 때문에, 역방향으로 진행할 때 어떤 위치에 있는 신호가 가장 강한 것이었는지 알 수가 없음 -> Switces 도입

Switces는 max pooling할 때, 가장 강한 자극을 가지고 있었던 위치 정보를 저장해 뒀, unpooling할 때 switces 정보를 활용하여 가장 강화 위치로 정확하게 찾아갈 수 있도록 도와줌



Visualizing Method: Rectification



ReLU의 역과정을 수행하게 됨

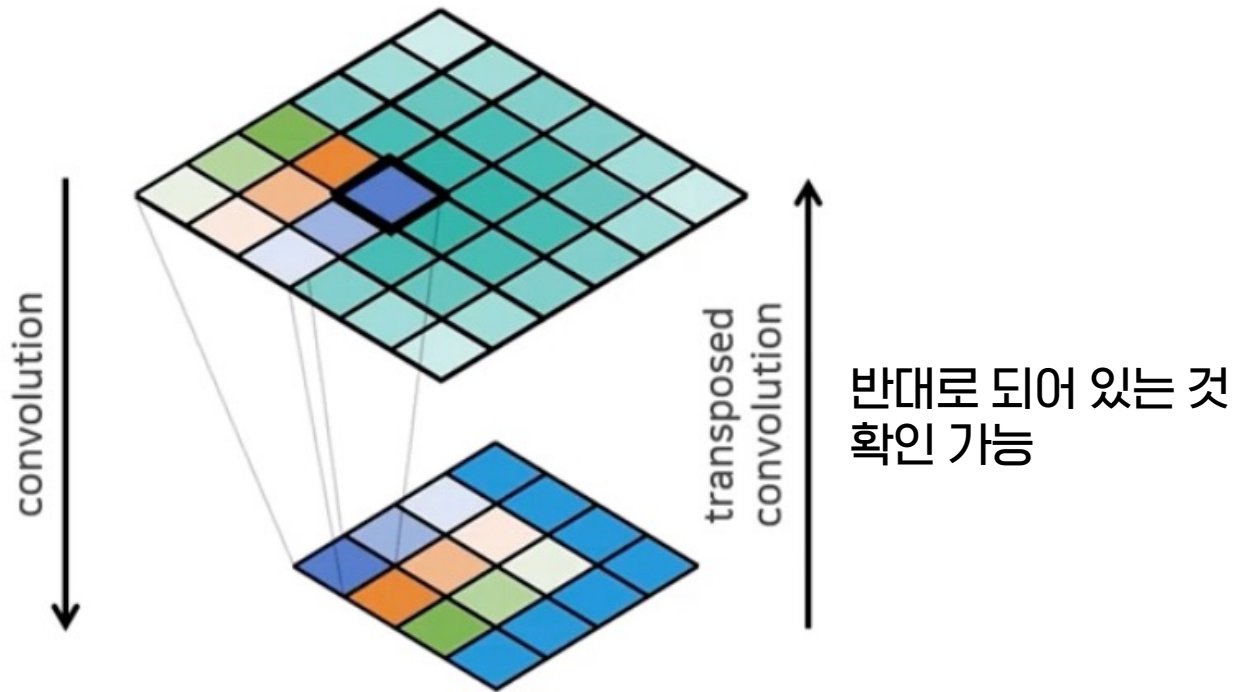
기존: 음의 값을 갖는 것은 0으로,
양의 값은 그대로 유지

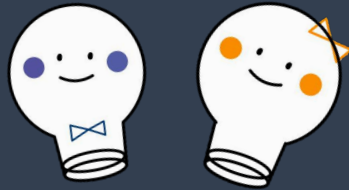
역과정: 음의 값을 갖는 것은 0으로,
양의 값은 그대로 유지

[양의 값은 문제가 없고, 음의 값들은 복원할
방법이 없어 0으로 유지 하였지만 문제가
없는 것으로 나타남]

Visualizing Method: Deconvolution

Filter에 대한 inverse 필터 연산 수행





우리만 따라와 Follow ADS

이상 ADS 황성아였습니다. 감사합니다.