

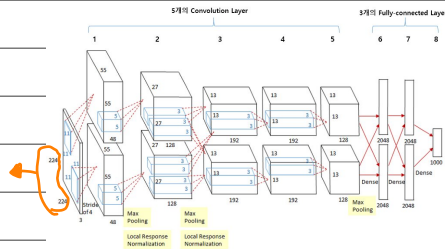
## AlexNet

### 참고

- 2012년 ILSVRC 대회 우승
- 저자의 이름을 따서 이름 생성

### 구조

- 2개의 GPU로 병렬 연산을 수행하기 위해 병렬적인 구조로 설계되었다.



논문에서 잘못표기해서

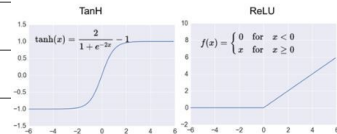
224 이고 원래 입력은 227이다.

- 8개의 레이어로 구성되어 있다. [5개의 컨볼루션 레이어, 3개의 f.c 레이어]
- 2, 4, 5번째 컨볼루션 레이어들은 일반적인 방식으로 진행되었지만,  
3번째 컨볼루션 레이어에서는 전 단계의 두 채널의 특성 맵들과 모두 연결되어 있다.

### 레이어 분석

#### ReLU

- LeNet-5에서 사용한 Tanh 함수 대신 ReLU를 사용하였다.
- 이같은 같은 정확도에 비해 속도가 6배 빨랐기 때문이다.

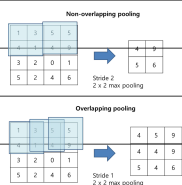


#### Drop-out

- 과적합을 막기 위해서 f.c Layer의 뉴런 중 일부를 생략하면서 학습전행
- 이작업은 훈련 할때만 적용되고, 테스트 할때는 다르고 진행한다.

#### Overlapping Pooling

- LeNet-5에서는 평균 풀링을 사용했지만 여기서는 최대 풀링을 사용하였다.
- stride를 2로 커널 사이즈보다 작게 설정 하는 것을 의미한다.

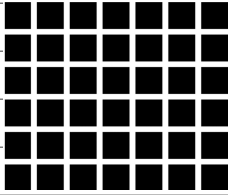


## Local Response Normalization (LRN)

• 아래 그림처럼 편차가 크면 아래처럼 (회색) 즉면 억제 현상 (로컬처링 강함 자극이 검정 부분에 둘러싸이면서 억제 당하는 현상)

이러한 현상 처럼 너무 큰자 특출난 feature는 다른 feature에 영향을 준다.

• 위와 같은 문제로 정규화 시키는 방법이 LRN이다.



• LRN 공식은 아래와 같으며 벡터들 간에 간섭이 없다.

• 그와는 반대로 BN같은 경우 파카 벡터로 서로 간섭을 한다.

## LRN

$$\hat{b}_{x,y}^i = a_{x,y}^i / (k + \alpha \sum_{j=\max(0, i-n/2)}^{\min(N-1, i+n/2)} a_{x,y}^j)^{\beta}$$

where  
 $\hat{b}_{x,y}^i$  — regularized output for kernel  $i$  at position  $x, y$   
 $a_{x,y}^i$  — source output of kernel  $i$  applied at position  $x, y$   
 $N$  — total number of kernels  
 $n$  — size of the normalization neighbourhood  
 $\alpha, \beta, k, (n)$  — hyperparameters

• 예제 1

- Feature Map 1 (10, 10, 10)
- Feature Map 2 (10, 10, 10)
- Feature Map 3 (10, 10, 10)

• 예제 2

- Feature Map 1 (10, 10, 10)
- Feature Map 2 (10, 10, 10)
- Feature Map 3 (10, 10, 10)
- Feature Map 4 (10, 10, 10)

2D LRN (LRN)을 위해 다음과 같이 데이터를 사용하십시오 (1D)

- $k = 1$
- $\alpha = 0.01$
- $\beta = 0.5$
- $n = 2$

예제 1에 대한 LRN

예제 2에 대한 LRN

2D LRN (LRN)을 위해 다음과 같이 데이터를 사용하십시오 (2D)

예제 1에 대한 LRN

예제 2에 대한 LRN

## BN

• 예제 1

- Feature Map 1 (10, 10, 10)
- Feature Map 2 (10, 10, 10)
- Feature Map 3 (10, 10, 10)

• 예제 2

- Feature Map 1 (10, 10, 10)
- Feature Map 2 (10, 10, 10)
- Feature Map 3 (10, 10, 10)
- Feature Map 4 (10, 10, 10)

예제 1에 대한 BN

예제 2에 대한 BN

2D BN (BN)을 위해 다음과 같이 데이터를 사용하십시오 (1D)

예제 1에 대한 BN

예제 2에 대한 BN

2D BN (BN)을 위해 다음과 같이 데이터를 사용하십시오 (2D)

예제 1에 대한 BN

예제 2에 대한 BN

## 데이터

• 과적합을 막기 위해 Drop out 말고도 data augmentation 이란 방법을 통해 데이터 양을 늘렸다.

• 총 2가지로 '수평반전 후 랜덤 crop', 'RGB 픽셀 값을 변환'

• Top-1 에러율을 1% 낮췄었다.

