

# AlexNet

## 0. Abstract, 2. The Dataset

1. 0.1% : ImageNet LSVRC-2010 의 1.2012 / 1000 labels

2012

2. 7% : top-1 error 17.5% (모델이 가장 확률이 높다고 예측하는  
- 5 11% (클래스가 정답이 아님 비율)  
(모델이 가장 높은 확률을 예측한 클래스는  
정답이 맞은 비율)

top5 15.3%

3. 구조(E1): 6 층 딥러닝

구조: 6 층 딥러닝

Conv + maxpool: 5 층

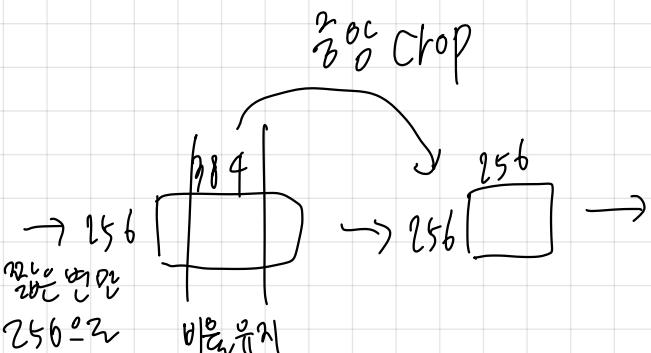
fc: 774 (Dropout + softmax)

activation: ReLU

func

Dropout

0.1%: 256x256  $\Rightarrow$  400



$$M_R = \frac{(R_1 + R_2)}{2} \Rightarrow R'_1 = R_1 - M_R, R'_2 = R_2 - M_R$$

$$M_G$$

$$M_B$$

$R_1, G_1, B_1$      $R_2, G_2, B_2$



224x224x3  $\Rightarrow$  "224x224"

( $\Rightarrow$  4.1

Data Augmentation)

## 1. Introduction

1. 학습: ImageNet  $\downarrow$   $\rightarrow$  원하는 모양  
정답은 ImageNet

$\rightarrow$  학습: 좋은 learning capacity X

but, CNN:  $H \times W$ 로 용량 제한 흡수할 가능성이

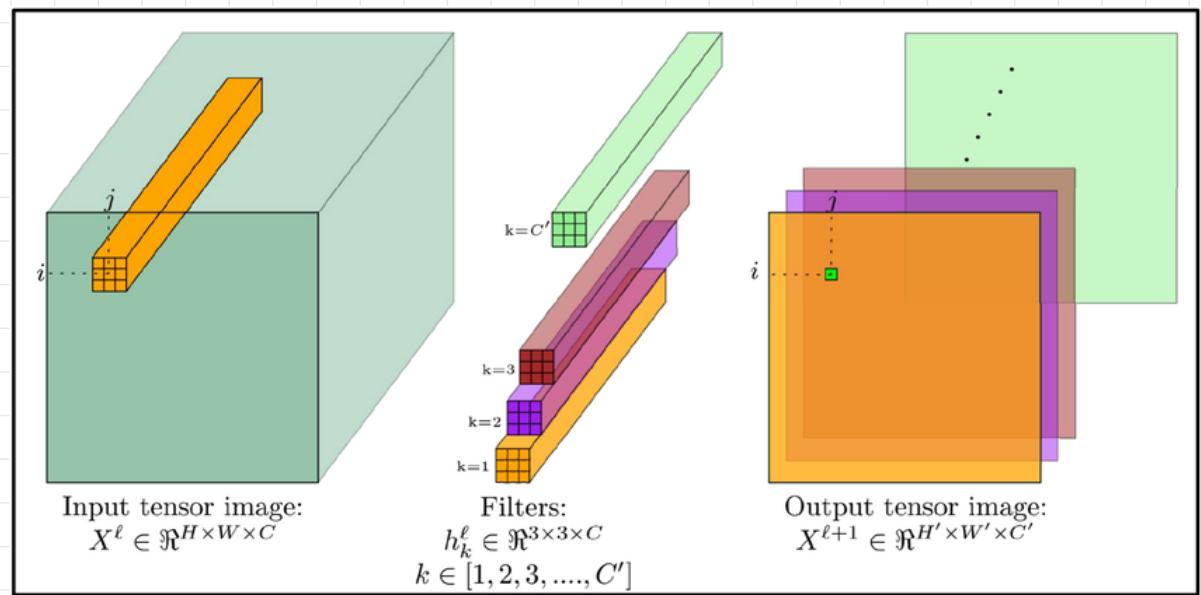
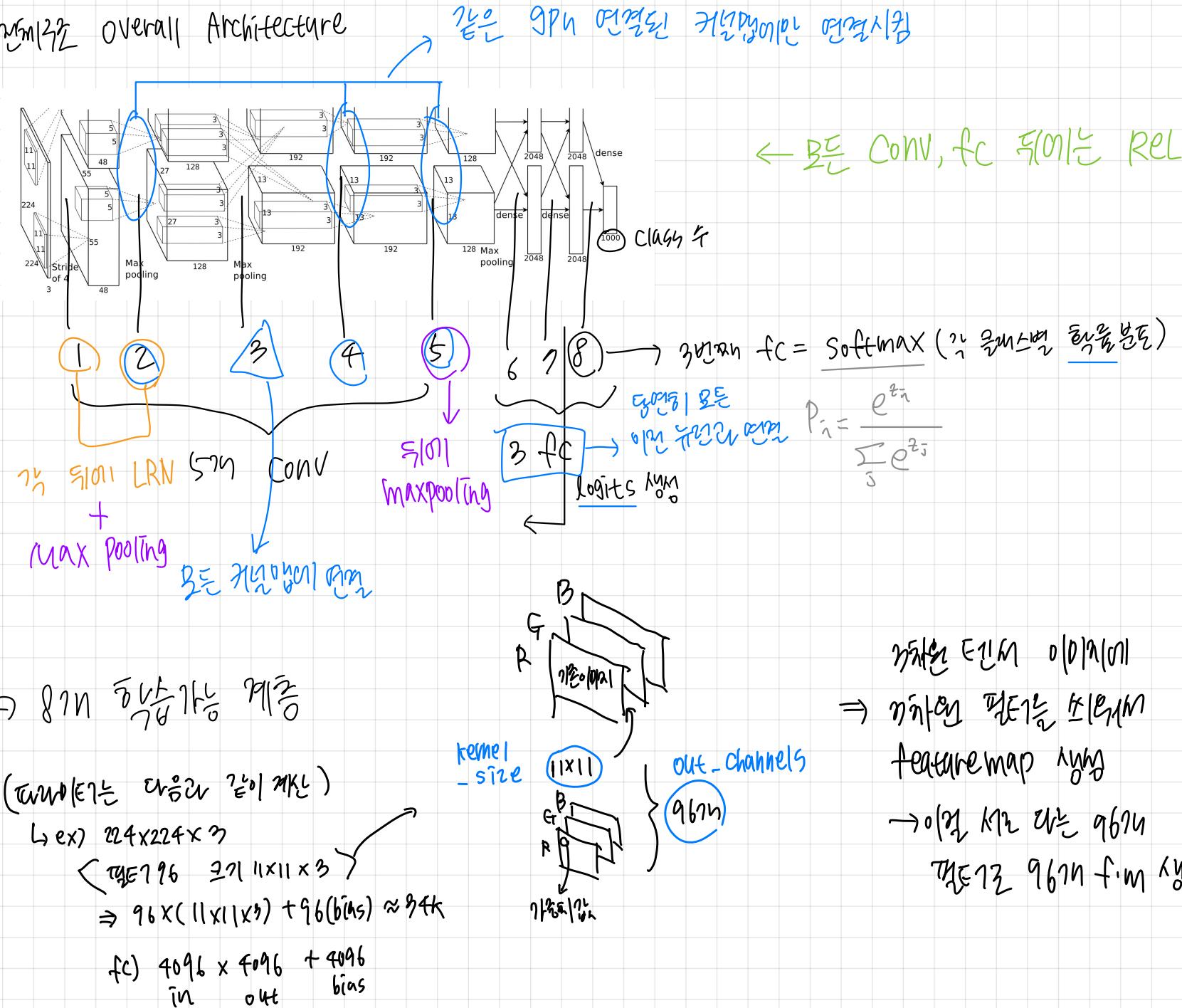
온바른 성질 (stationarity of statistics) : 같은 위치에 있는 픽셀이 다른 위치와 X

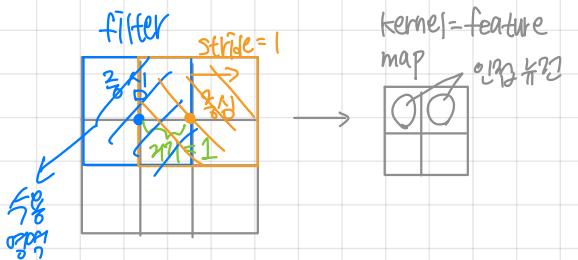
& locality of pixel dependency) : 같은 위치에 있는 픽셀들이 같은 특성을  
갖고 있는 특성을 갖는다

만약 fc의 위치를 전환할 수 있으면 수  $\downarrow$ , 성능  $\uparrow$

### 3. The Architecture

#### 1. Overall Architecture





→ 커널을 통해 이미지 뉴런간  
각 수용 영역의 중심은 차이

총 97 차원

	Layer	Img/ f.m	size	kernel_size	out_channels	stride	padding	
150528	Input	224x224x3						
254460	conv1	55x55x96	11x11x3	11	96	4	2	+LRN +ReLU
	max pooling	27x27x96	3x3x96			2		+ReLU
186624	conv2	27x27x256	5x5x96	256		1	2	+LRN +ReLU
	max pooling	13x13x256	3x3x256			2		+ReLU
64896	conv3	13x13x384	3x3x256	384		1	1	+ReLU
64896	conv4	13x13x384	3x3x384	384		1	1	+ReLU
43264	conv5	13x13x256	3x3x384	256		1	1	+ReLU
	max pooling	6x6x256	3x3x256			2		+ReLU

flatten  $6 \times 6 \times 256 = 9216$

4096	fc1	$9216 \rightarrow 4096$	+ Dropout
4096	fc2	$4096 \rightarrow 4096$	+ Dropout
num_classes	fc3	$4096 \rightarrow \text{num\_classes}$	

→ 총 97 차원 ;  $224 \times 224 \times 3 = 150528$

2. ReLU (Rectified Linear Unit):  $\max(0, x)$

→ 정사각형 시 tanh, sigmoid 등 풀려는 비선형은 속도 느림 //  $\oplus$   $|\tanh(x)|$  but, 과적합 방지 해결일본  
이유는  $\tanh(x)$ 은 X

3. Local Response Normalization(지소 응답 정규화) : LRN // ⊕ LCN(지소 and 정규화)

→ Rel. V 는 풍부한 자료, 유익한 의견이나 제언 필요 X (Positive 의견이나 제언 가능)

→ ReLU ဆုံးစွဲ local normalization ဆုံးမြတ်စွာ ပို၏

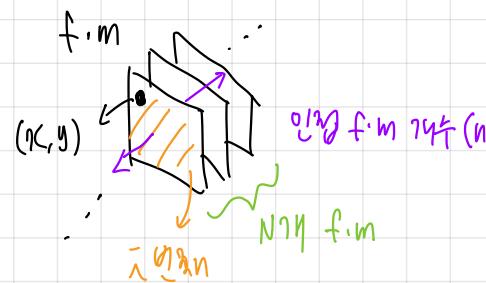
$$b_{x,y}^i = \left( \Delta_k + \Delta \sum_{j=\max(0, i-n/2)}^{\min(N-1, i+n/2)} (a_{x,y}^j)^2 \right)^{\frac{1}{2}}$$

▷; hyper-parameter

$k=2$	$n=5$	$\alpha=10^{-4}$	$\beta=0.15$	$\lambda_0^{lm}$
-------	-------	------------------	--------------	------------------

ReLU 후의 N개 f.m % 확률

$$\Rightarrow \begin{matrix} \text{top 1} & 1.4\% \downarrow \\ \text{top 5} & 1.2\% \downarrow \end{matrix}$$



#### 4. Overlapping Pooling (겹첩 풀링)

→ Pooling : 중복된 패턴들(커널 맵)의 같은 뉴런들에 대해 한 번에 몇몇 뉴런들의 출력을 얻기,  
같은 패턴마다 어떤

이동 폴딩이 예측하는 영역이  $z \times z$  만으로서  $\frac{\text{stride}}{\text{영역의 이동}} = \frac{z(\text{원도구 크기})}{\text{한 번에 보는 크기}}$  (z by z) 를 동일하게

$\rightarrow \text{Syn}: S \subset T$

↳  $s=2, l=2$  (4중입) 일 때와 동일하게,

top 1 0,4% ✓

top 5 0.3% ↓

과적행↓

## 4. Reducing overfitting (과적합 감소)

## ① label-preserving transformation 27/21

## 1. translation + horizontal reflection

$256 \times 256$

↳ transforms, Random Horizontal Flip( $p$ ): 전체 훈련 데이터셋을  $p$  확률 만 수평반전 (좌우반전)

224 × 224 [1/2] 무지개위 추출

(transforms, RandomCrop(224))

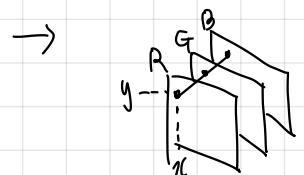
→ test 시에로 사용: 각 원본당  $224 \times 224$  터치 574 + 각 칸의 번호  $\Rightarrow$  각 총 1074 터치

$$\begin{array}{r} 2121471 \\ + 2606171 \\ \hline \end{array}$$

각 원소에 대한 softmax 후 정규화

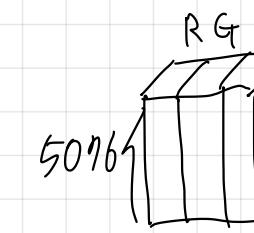
2. RGB 채널 강도 조정  $\rightarrow$  top 1% 흐트

↳ 물체 정체성 (Identity)이 조영 강도와 상관이 있음을 발견



Input img

$$\Rightarrow I_{xy} = \begin{bmatrix} I_{xy}^R \\ I_{xy}^G \\ I_{xy}^B \end{bmatrix} \text{ 가 } 224 \times 224 \text{ 인 행렬 } (\text{1장 이미지}) \rightarrow N\text{장 이미지} \rightarrow (50/176 \times N) \times 3 \text{ (2차원 행렬)}$$



N장 : (제작 50/176 장) 50/176장까지 이어붙임)

↓ 공분산 행렬

1) “채널별로”  $M_R, M_G, M_B$  구하고 각 열별 원소들에서 배수하기  $\Rightarrow X$  행렬

$$2) \frac{1}{N} X^T \cdot X = [3 \times (50/176 \times N)] \times [(50/176 \times N) \times 3] \} = 3 \times 3 \text{ 공분산 행렬} = \Sigma$$

3) PCA 수행:  $P$ (고유벡터),  $\lambda$ (고유값)

↳  $\Sigma P = \lambda P$  수행 ( $\det(\Sigma - \lambda I) = 0$  해면  $\lambda_1, \lambda_2, \lambda_3$  찾고  $\Sigma P = \lambda P$  해기)

↓

$[P_1 \ P_2 \ P_3]$   $\begin{bmatrix} \alpha_1 \lambda_1 \\ \alpha_2 \lambda_2 \\ \alpha_3 \lambda_3 \end{bmatrix}$  의 값은 모든 각 픽셀에 대해서

$$\alpha = \frac{\text{가장 큰 값으로서 봄}}{\mu=0, \sigma=0.1}$$

② dropout: 순전파 시 일부 뉴런 끄기 (역전파도 영향 미사용)

$\rightarrow$  1) 뉴런끼리 차이난 모든 뉴런, 어떤 다른 그룹의 (끼끼) 네트워크 = 암상수 (단, 가중치는 같은 연결이거나 다른 네트워크에서 공유) / 2) 수령까지 반복 횟수 증가

$\rightarrow$  test: 모든 뉴런을 쓰되, 출발값에  $\times 0.5 \rightarrow$  암상수의 (기하) 평균 확장하기 위해

## 5. Details of learning

1. Optimizer: SGD

$\rightarrow \text{optim.SGD(model.parameters(), lr, momentum, weight_decay)}$

$\rightarrow \text{momentum: 0.9 / weight-decay: 0.0005}$

$\hookrightarrow$  "관성": GD의 가속을

증상에서 진동하는 문제

해결하기 위한 이전 기울기의

이동 방향 기억해서 전환

기울기에서 더러운 업데이트

(이전 방향대로 업데이트)

$\hookrightarrow$  과적합 막기 위한 Regularization (Weight 치는 것 막기)

:  $\text{loss}(y, \hat{y}) + \lambda \sum_i w_i^2$  ( $\text{loss}$  줄이는 것 목표니까)

(training errors 줄이듯)

$\Rightarrow$  최종 weight update 규칙:  $w_{\bar{n}+1} = w_{\bar{n}} + v_{\bar{n}+1}$

$$= 0.9 \cdot v_{\bar{n}} - 0.0005 \cdot \epsilon \cdot w_{\bar{n}} - \epsilon \cdot \left\langle \frac{\partial L}{\partial w} \Big|_{w_{\bar{n}}} \right\rangle_{D_{\bar{n}}}$$

$\hookrightarrow \bar{n}$ : iteration

$v_{\bar{n}}$ : momentum 변수

$\epsilon$ : lr

$D_{\bar{n}}$ :  $\bar{n}$  번째 미니배치

$E(x, y) \in \left\langle \frac{\partial L}{\partial w} \Big|_{w_{\bar{n}}} \right\rangle_{D_{\bar{n}}}$ ; 해당 미니배치 내부 손실함수 기울기

$\hookrightarrow$  가중치 초기화:  $M=0, \Gamma=0.01$  (가우시안)

$\rightarrow$  bias: conv2,4,5 + fc1,2,3 모두 1로 초기화해서  $y = w^T x + b = 1$

값이 0인 경우  $\text{ReLU}=0$ 이 되는 문제 ( $\text{ReLU}(\theta) = \begin{cases} \theta & \theta > 0 \\ 0 & \theta < 0 \end{cases}$ )를 막음

$\rightarrow$  학습 초기에 수렴 가능

( $w[2] = 0$ )

2. batch size: 128 / epochs = 90 / lr = 0.01 (학습률  $\times$  (plateau),  $\frac{1}{10}$ 씩 감소, 3번 감소)

## 6. Results

## 7. Discussion

→ depth 가 성능 단층에 핵심