ImageNet Classification with Deep Convolutional Neural Networks (NIPS 2012)

AlexNet

Introduction

- Object recognition의 성능을 올리려면
  - 더 큰 데이터셋
  - 더 강력한 모델
  - Overfitting을 막기 위한 방법 사용

- 현재는 작고 간단한 데이터셋에서만 좋은 결과 (ex: MNIST)

- 더 크고 label 종류가 많은 데이터셋(ImageNet)을 학습하기 위해선 CNN
  - 크고 조절가능한 learning capacity
  - Prior knowledge
  - FFNN보다 쉬운 학습

Introduction

- CNN은 상대적으로 학습이 쉽지만, 여전히 high-resolution image에 적용하기는 힘들다
  - GPU
  - ImageNet이라는 큰 데이터셋 등장 => overfit 없이 학습 가능

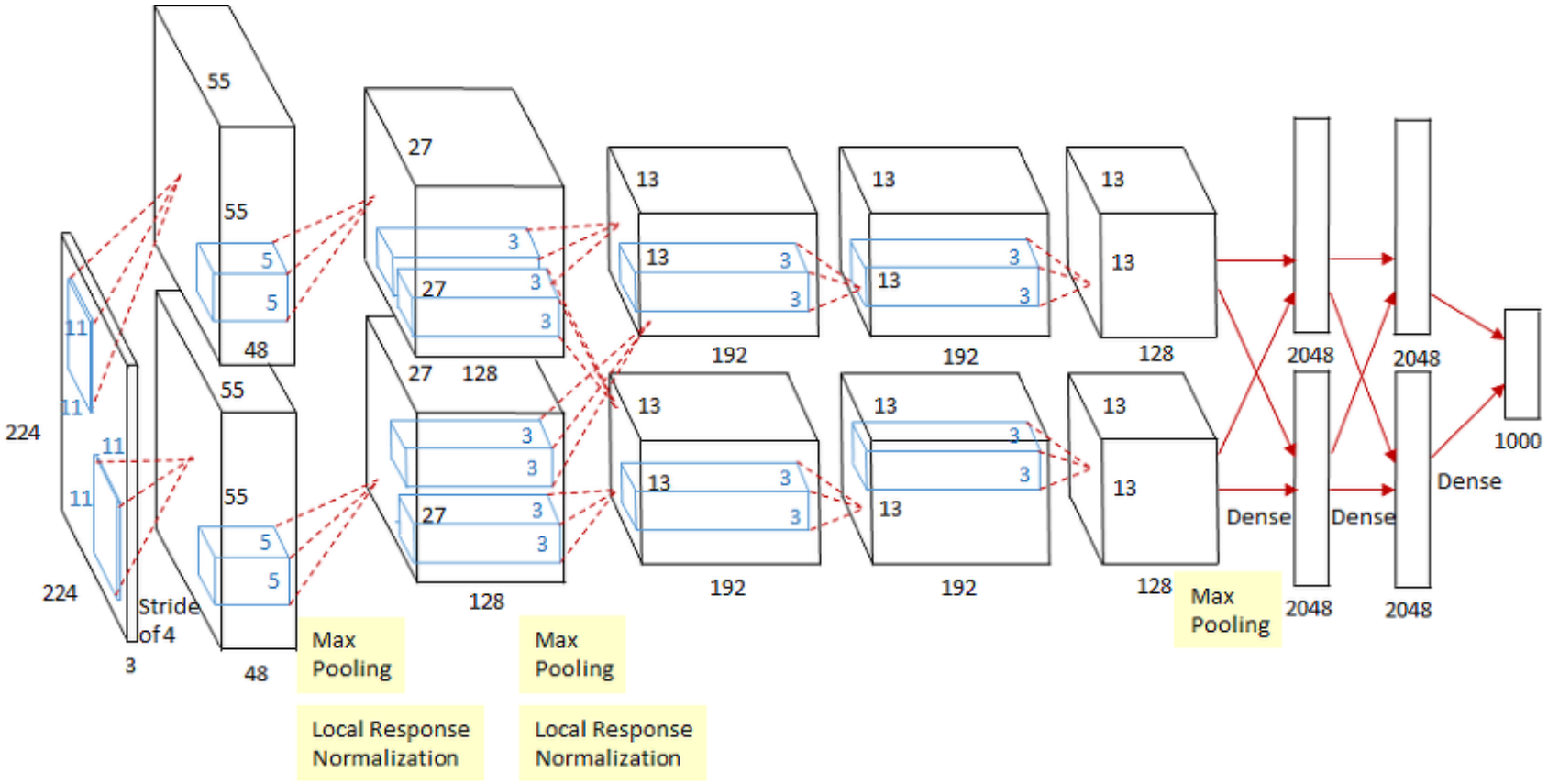
- 여러가지 기술들 조합해서 큰 CNN 모델을 성공적으로 학습시킴
  - ILSVRC-2012 대회에서 압도적인 우승
  - 이전까지 딱히 성과가 없던 컴비전 딥러닝 분야에 큰 이슈
  - GPU가 더 발전하면 네트워크 사이즈를 더 늘릴 수도 있을 것

Dataset - ImageNet

- 1500만 개의 high-resolution colored images
- 22000개의 카테고리 (label)
- Resolution이 이미지마다 달라서 256 x 256으로 고정해서 사용

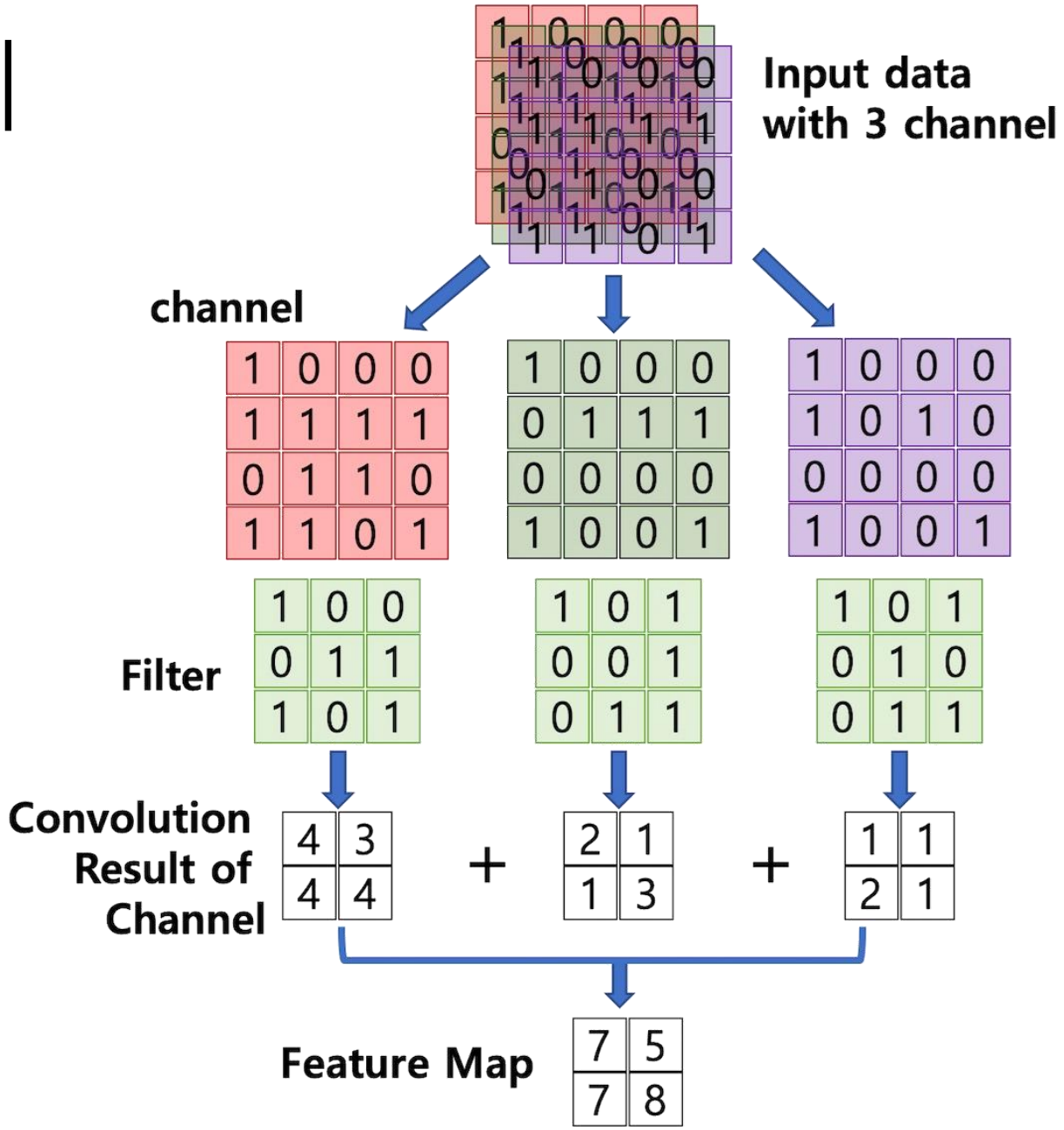
Architecture

- 5 CNN layers
- 3 Fully Connected layers



+) CNN 용어 다시보기

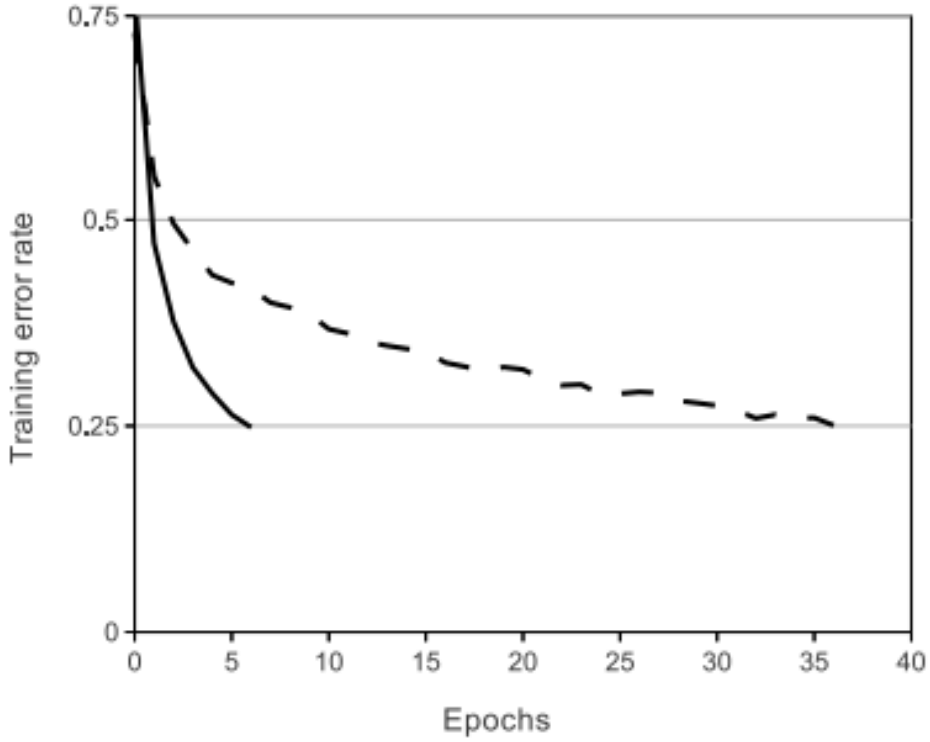
- Kernel == filter
- Feature map
- pooling layer



Architecture – ReLU Nonlinearity

- 활성화함수를 뭘 쓸까
  - Tanh
  - Sigmoid
  - ReLU = max(0,x)

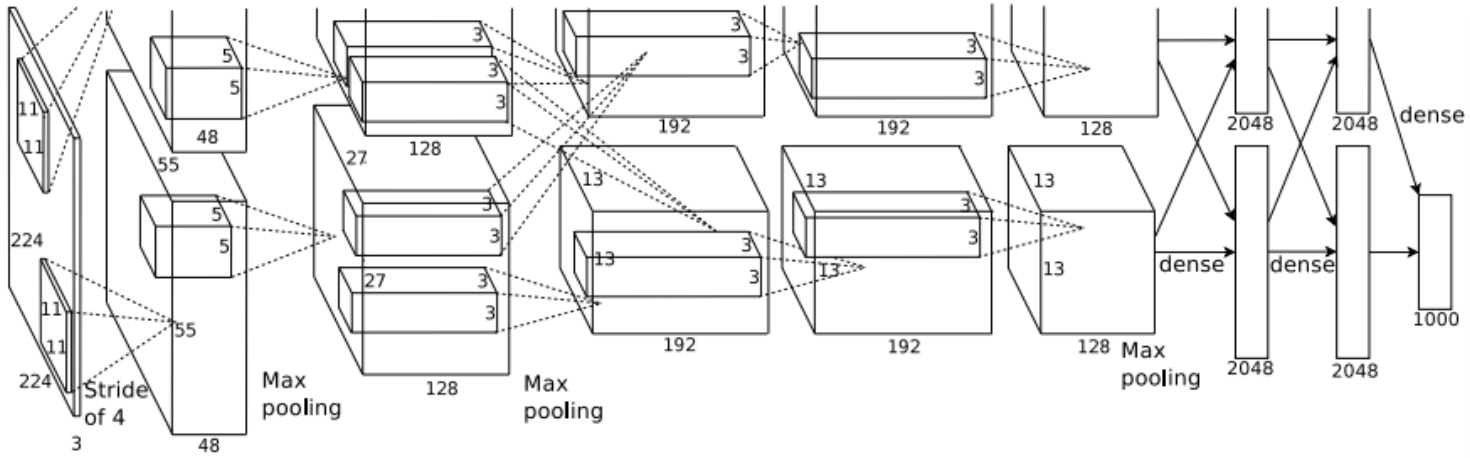
- ReLU
  - Non-saturating
  - 더 빠른 학습



- 빠른 학습은 큰 데이터셋으로 큰 모델을 학습시킬 때 성능에 큰 영향을 줌

Architecture – Multiple GPUs

- GPU 메모리가 부족해서 두 개를 쓰기로함
- Kernel(뉴런)을 반씩 GPU에 넣음
- 몇 개의 레이어에서만 서로 간섭함
  - Computation 용량 조절

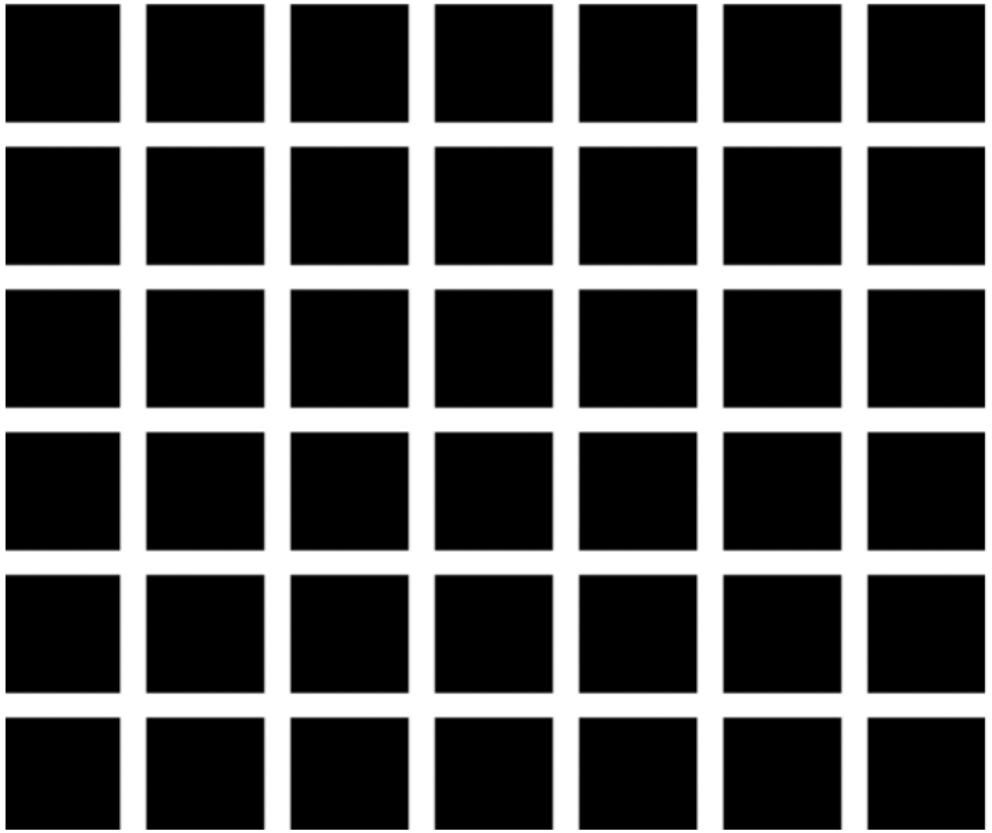


Architecture – Local Normalization

- ReLU 덕분에 saturation 걱정은 없음
- 하지만 여전히 local Normalization은 Generalization을 도움

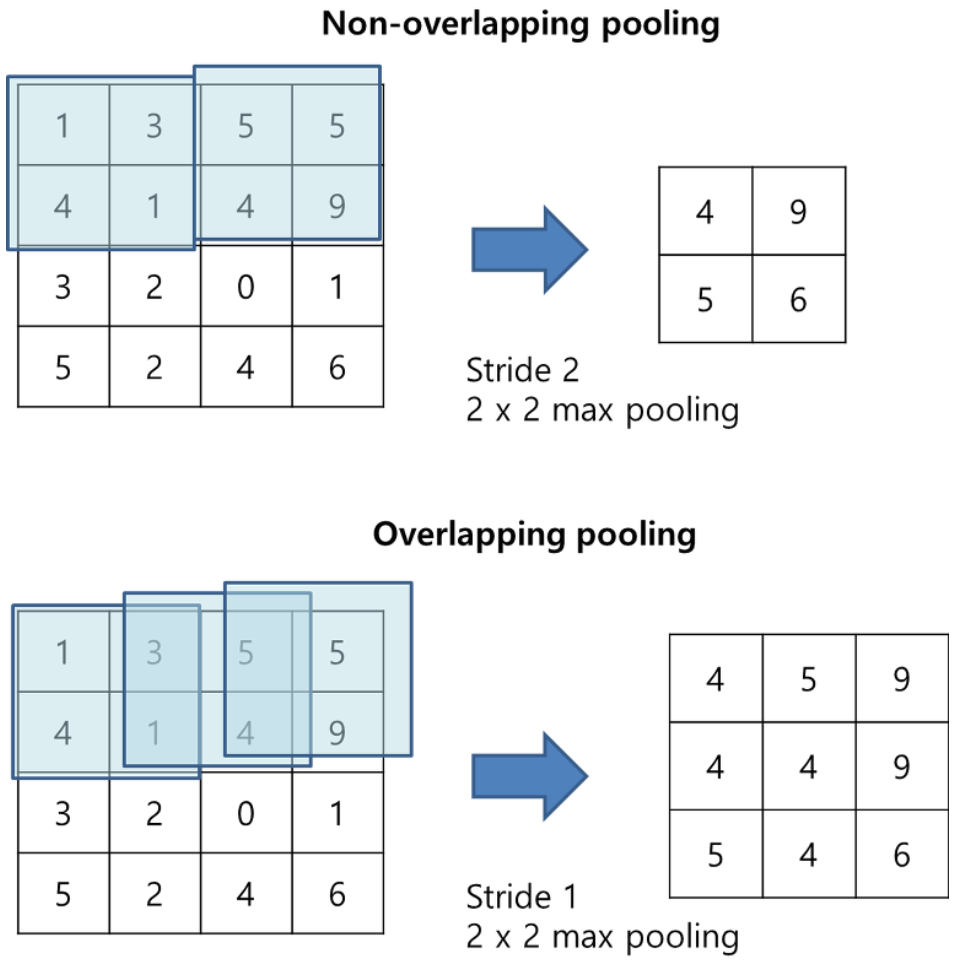
$$b^i_{x,y} = a^i_{x,y} / (k + \alpha \sum_{j=max(0,i-n/2)}^{j=min(N-1,i+n/2)} {a^j_{x,y}}^2)^\beta$$

where
$b^i_{x,y}$ – regularized output for kernel i at position x, y
$a^i_{x,y}$ – source ouput of kernel i applied at position x, y
N – total number of kernels
n – size of the normalization neigbourhood
$\alpha, \beta, k, (n)$ – hyperparameters



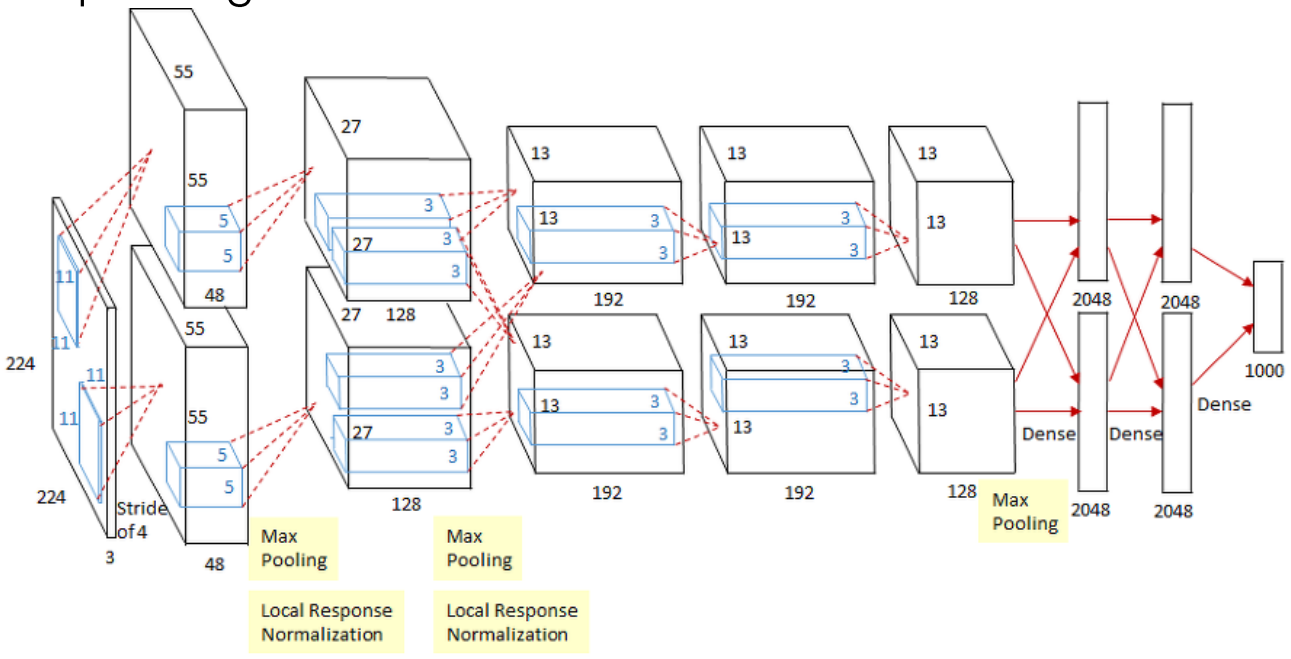
Architecture – Overlapping pooling

- Overlapping max pooling
- Error rate 감소
- 살짝의 overfit 방지



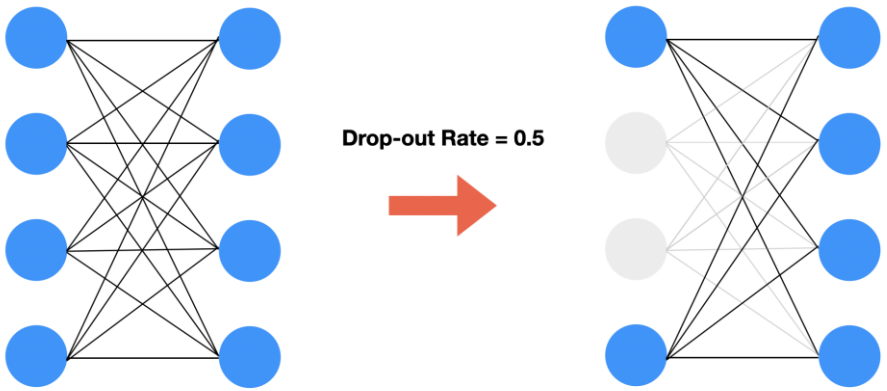
Architecture

- 5 CNN -> 3 FFNN -> 1000-way softmax
- Object 함수: multinomial logistic regression
- 2, 4, 5번째 conv layer는 GPU 따로
- 1, 2번째 conv layer는 Response-normalization
- 1, 2, 5번째 conv layer는 (Overlapping) Max pooling
- 모든 layer에는 활성화함수로 ReLU를 사용



Reducing Overfitting

- 6000만 개의 파라미터
- 1000개 클라스의 ILSVRC 데이터로는 overfitting
- 두 가지 방법으로 해결
  - Data Augmentation
    - Image translation, horizontal reflections
    - RGB 조정 – PCA를 통해
  - Dropout



https://heytech.tistory.com/

Details of Learning

- 최적화 방법: stochastic gradient descent
- Batch size: 128
- Momentum: 0.9
- Weight decay: 0.0005
- Weight은 정규분포로 초기화 (평균 0, 표준편차 0.01)
- Bias는 몇 개 레이어는 1, 나머지는 0으로 초기화
- Learning rate: 0.01 (validation error가 줄지 않으면 조금씩 바꿈)
- 90 epochs
- 120만 이미지
- 5, 6일 소요

$$v_{i+1} := 0.9 \cdot v_i - 0.0005 \cdot \epsilon \cdot w_i - \epsilon \cdot \left\langle \frac{\partial L}{\partial w}\Big|_{w_i} \right\rangle_{D_i}$$

$$w_{i+1} := w_i + v_{i+1}$$

Results

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SIFT + FVs [7]	—	—	*26.2%*
1 CNN	40.7%	18.2%	—
5 CNNs	38.1%	16.4%	**16.4%**
1 CNN*	39.0%	16.6%	—
7 CNNs*	36.7%	15.4%	**15.3%**

Table 2: Comparison of error rates on ILSVRC-2012 validation and test sets. In *italics* are best results achieved by others. Models with an asterisk* were "pre-trained" to classify the entire ImageNet 2011 Fall release. See Section 6 for details.

Qualitative Evaluations

Qualitative Evaluations

Discuss

- 크고 깊은 CNN을 지도학습으로 학습시키는 것만으로도 최고 기록
- 아키텍처에서 어떠한 Conv 레이어를 없애더라도 성능이 크게 안좋아짐
- 컴퓨팅 파워가 좋아지면 pre-training을 사용해도 좋을 듯

질문