

AlexNet

ImageNet Classification with Deep Convolutional Neural Networks

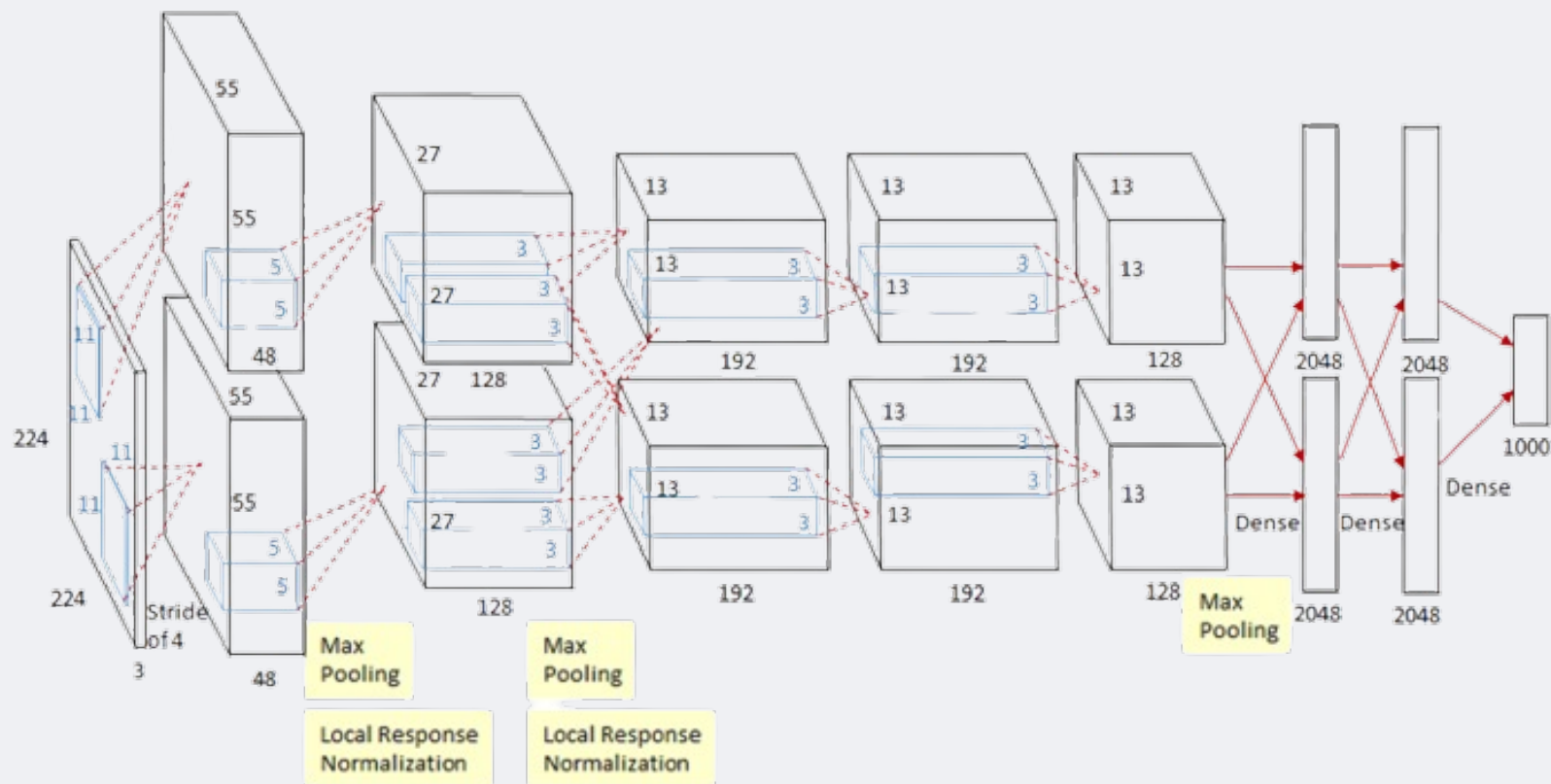
23.11.06.
김한주, 윤지희, 이승민

목차

1. AlexNet Contribution
2. AlexNet 구현
3. AlexNet 결과 정리

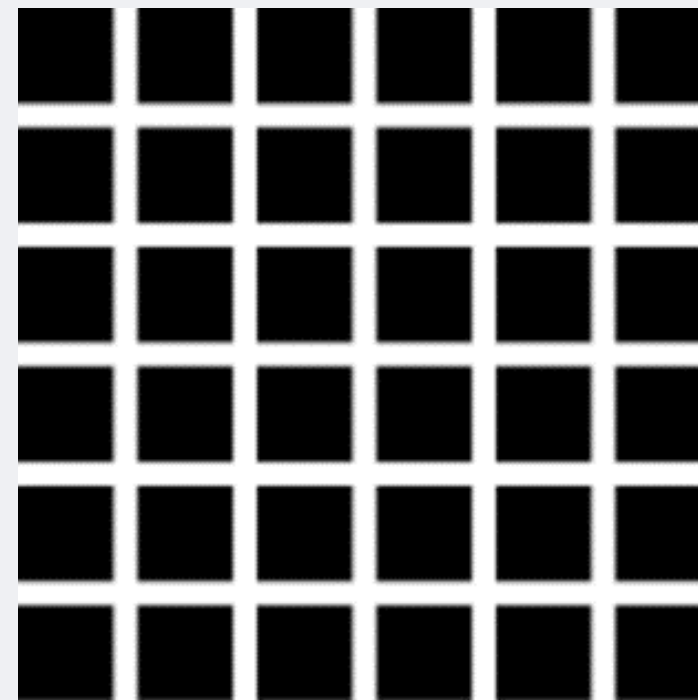
Contribution(1) - 27개의 GPU

2개의 GPU 병렬 연산 방식(LeNet-5와의 차이점)



Contribution(2) - ReLu, LRN

이전 TanH방식과 다른, ReLu를 사용해 unsaturation, non-linear 방식으로 6배의 속도 향상 및 깊은 학습 가능



Relu의 값을 그대로 사용하지 않고 LRN을 통해서 generalization

LRN에 영향을 준 ‘측면억제효과’의 예

Contribution(3)

깊어진 레이어, Overfitting 해결을 위해

Overlapping Pooling

unoverlapping pooling 보다 overfitting 해결에 도움을 줌

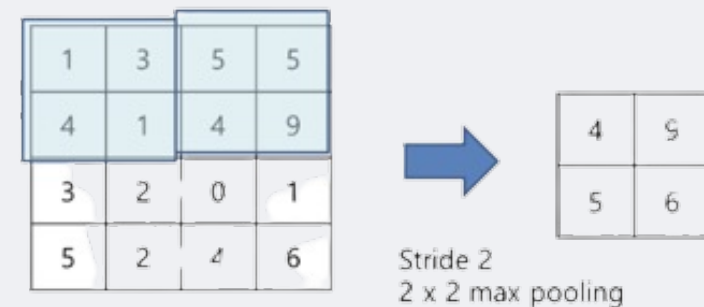
Data agumentation

: 수평반전(horizontal reflection) + Random Crops

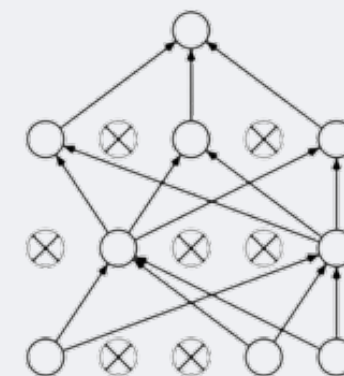
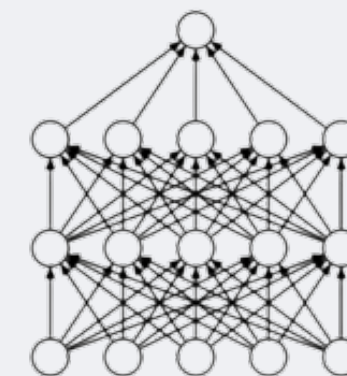
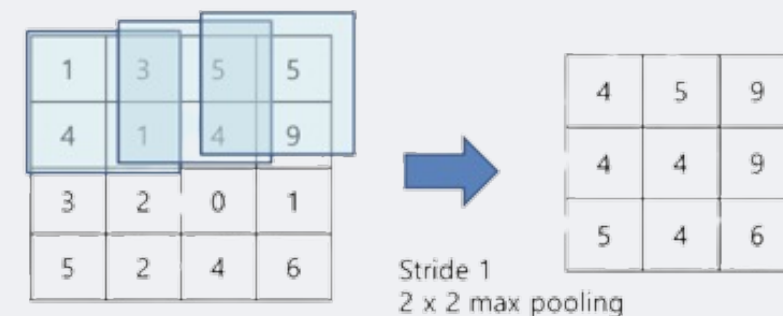
DropOut

: 앙상블 효과를 비슷하게 내기 위해서 dropout 방법 채택

Non-overlapping pooling



Overlapping pooling

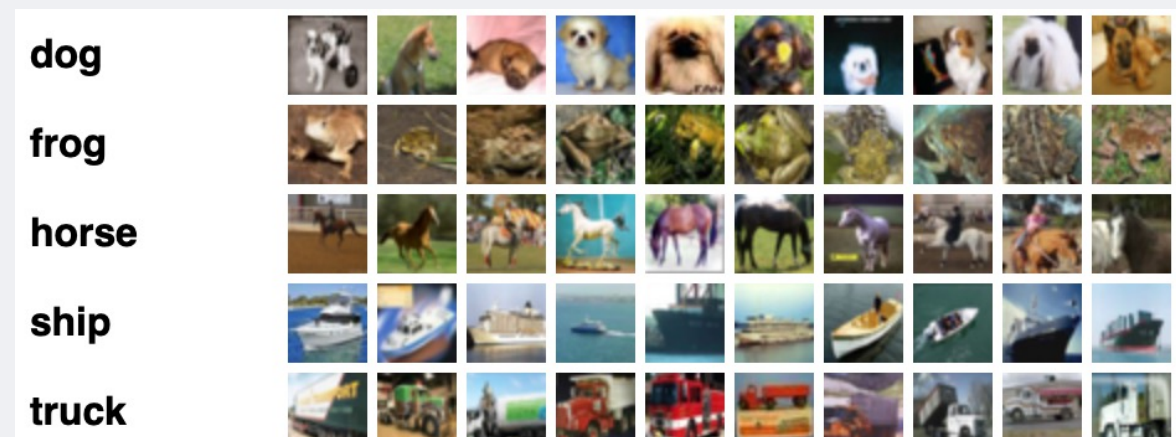
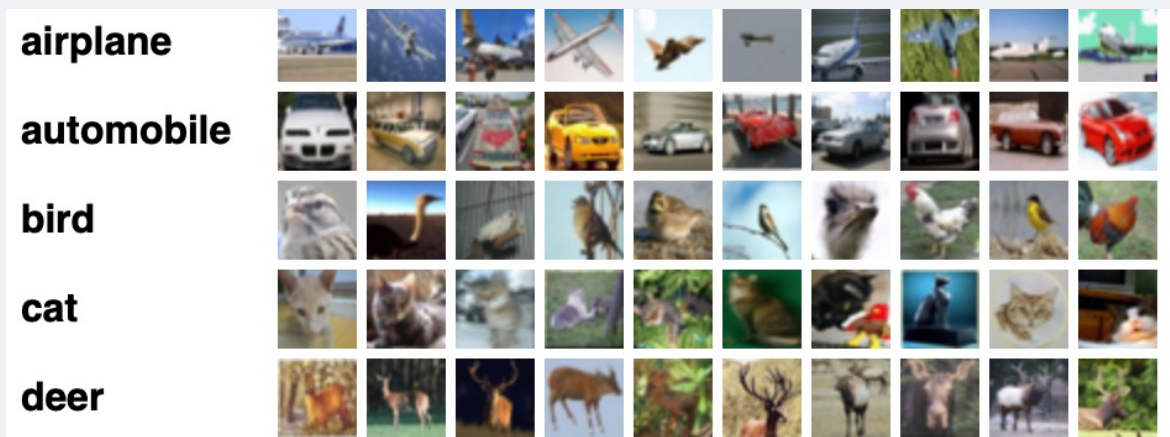


구현(1) - Dataset

Tensorflow 사용 : 익숙하지 않은 프레임워크를 사용해보기 위해서

1.Dataset

1. 기존 ImageNet은 용량이 너무 크기 때문에, 작은 CIFAR-10을 가져왔다.
2. airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck



구현(2) - Pipeline

2. Pipeline

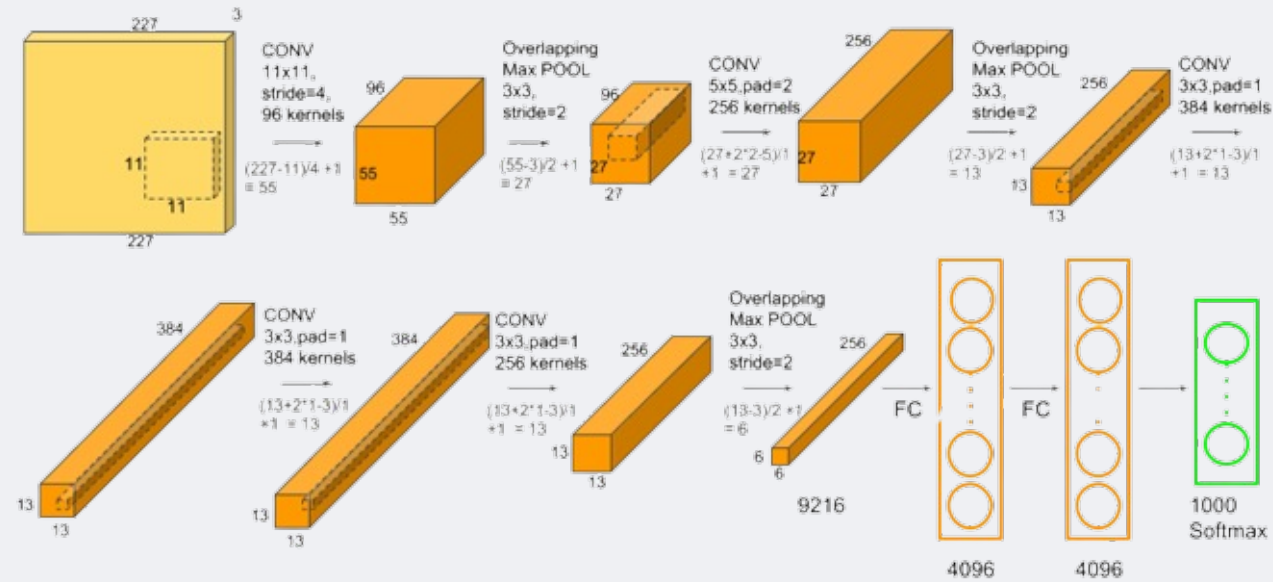
1. 데이터 전처리, 셔플, 배치 사이즈 만큼 뽑기
2. model 구조 alexnet에 맞춰서 작성

1.5*CONV(96 - 256 - 384 - 384 - 256)

1. 맨 처음 CONV를 제외하고 이미지 사이즈 변하지 않음, 이미지 사이즈는 MaxPooling
2. CONV layer 뒤에는 Batch Norm -> 논문의 LRN은 지금은 LRN을 사용안하기 때문에 batch를 사용

2.3*FC

1. 0.5 dropout rate
2. Softmax output



구현(3) - Fit, Evaluation

3. Fit

1. epochs = 10

4. Evaluation (해석 부분에 다름)

1. test 데이터로 evaluate 진행 → loss: 0.7798 – accuracy: 0.8114
2. 논문에 비해 성능이 낮은 이유는 적은 epochs 때문임.

결과

10 epochs 기준

구현(1) - loss 0.7798 - accuracy 0.8114

구현(2) - loss 0.8295 - accuracy 0.7125

논문과의 결과 차이: Smaller epochs

구현(1),(2) - LRN vs. BN (non-trainable vs trainable)

