

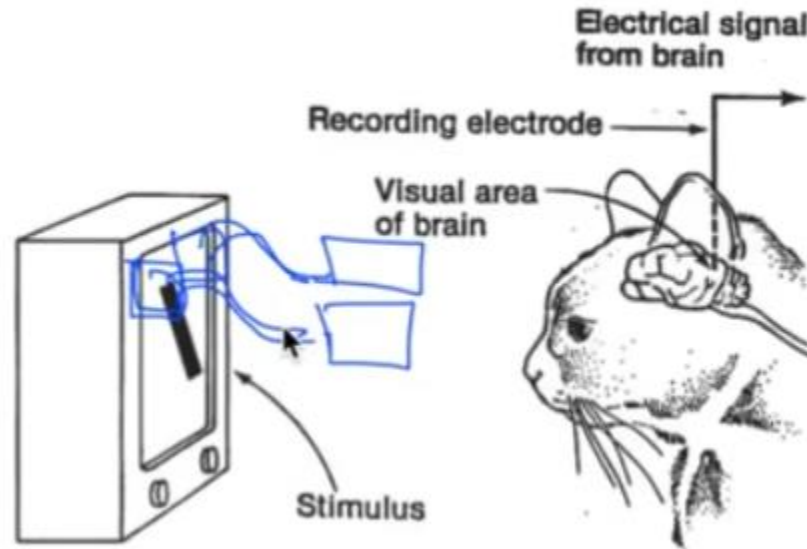
4주차 ML/DL 스터디 발표

GDSC Hanyang

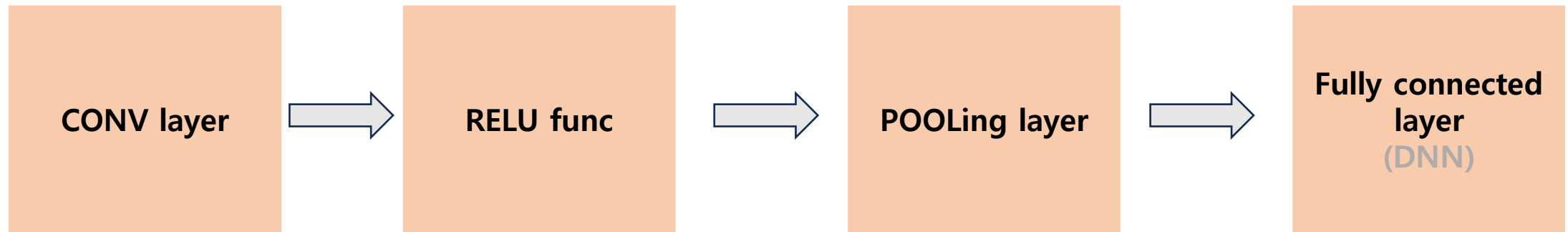
ML/DL core
김남호

CNN

- Convolutional Neural Network
- 이미지 분류, 객체 감지 및 이미지 처리 작업에 사용되는 딥 러닝 아키텍처

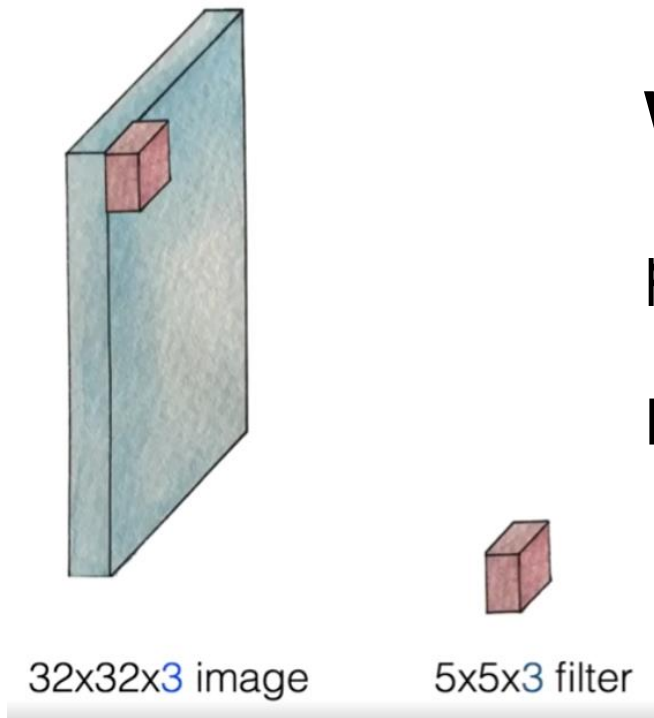


CNN



Convolution layer

- 합성곱 연산



$Wx+b$ (3차원 filter)

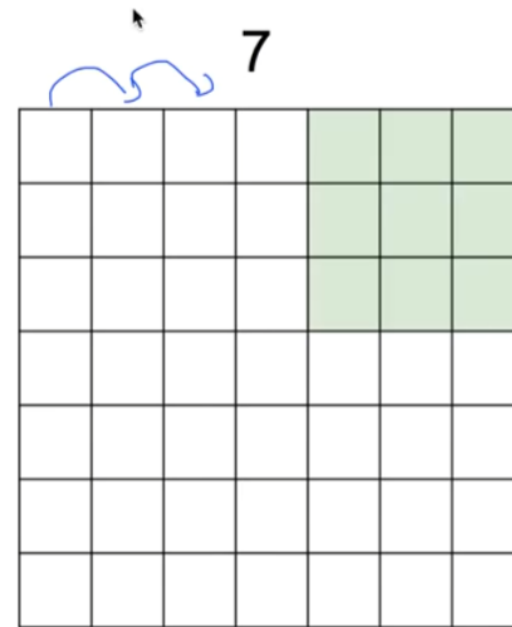
param 개수 : filter당(5x5x3) -CNN모델마다 다름

Linear regression => 하나의 값으로

Convolution layer

- Stride : 필터의 위치를 바꾸는 이동 거리

- Output size : $(N-F)/\text{Stride} + 1$



7x7 input (spatially)
assume 3x3 filter



=> 5x5 output

7

Padding

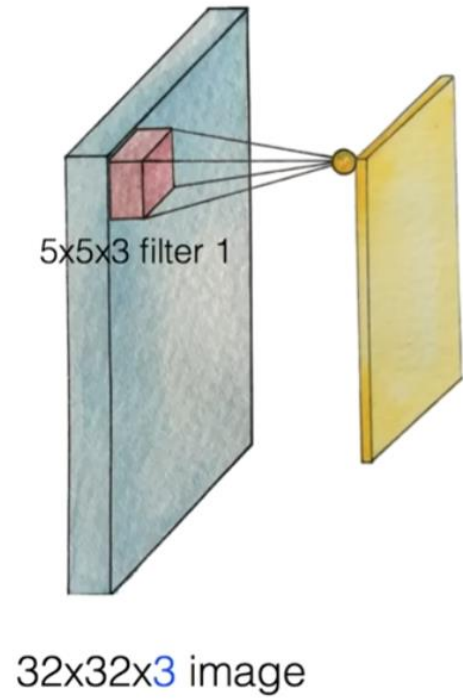
- Input : 7x7
- Filter : 3x3
- Pad with 1 pixel

=> **7x7 output**

데이터 손실x

0	0	0	0	0	0			
0								
0								
0								
0								

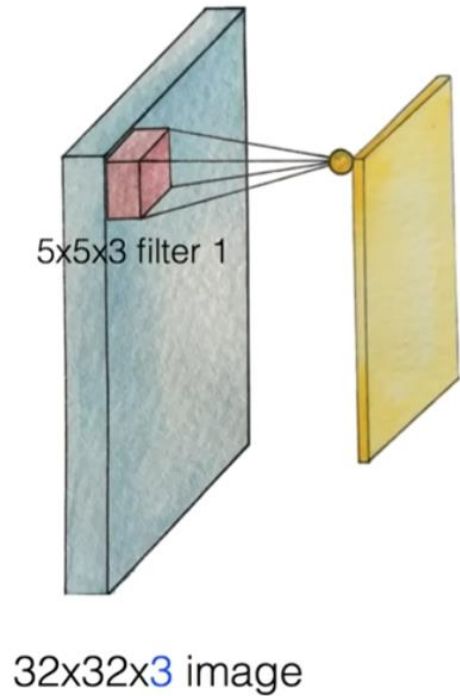
Convolution layer



6개 filter(5x5x3) -> conv layer 적용



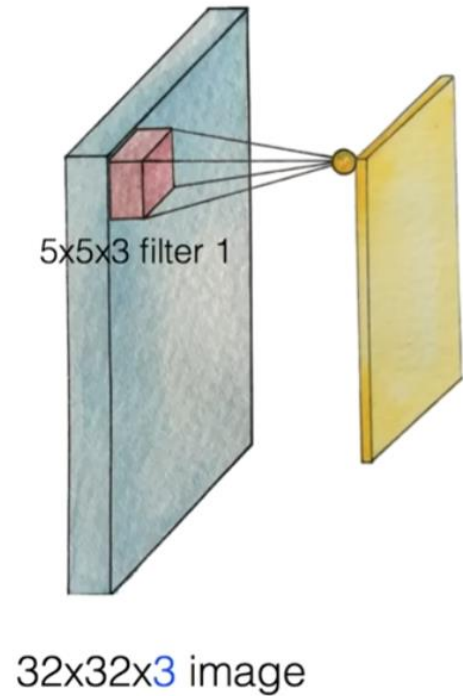
Convolution layer



6개 filter(5x5x3) -> conv layer 적용



Convolution layer



6개 filter(5x5x3) -> conv layer 적용

Relu func도 같이 적용



Pooling layer

- CNN의 conv layer 증가 -> 고차원, 多 param

=> overfitting 등의 문제 발생

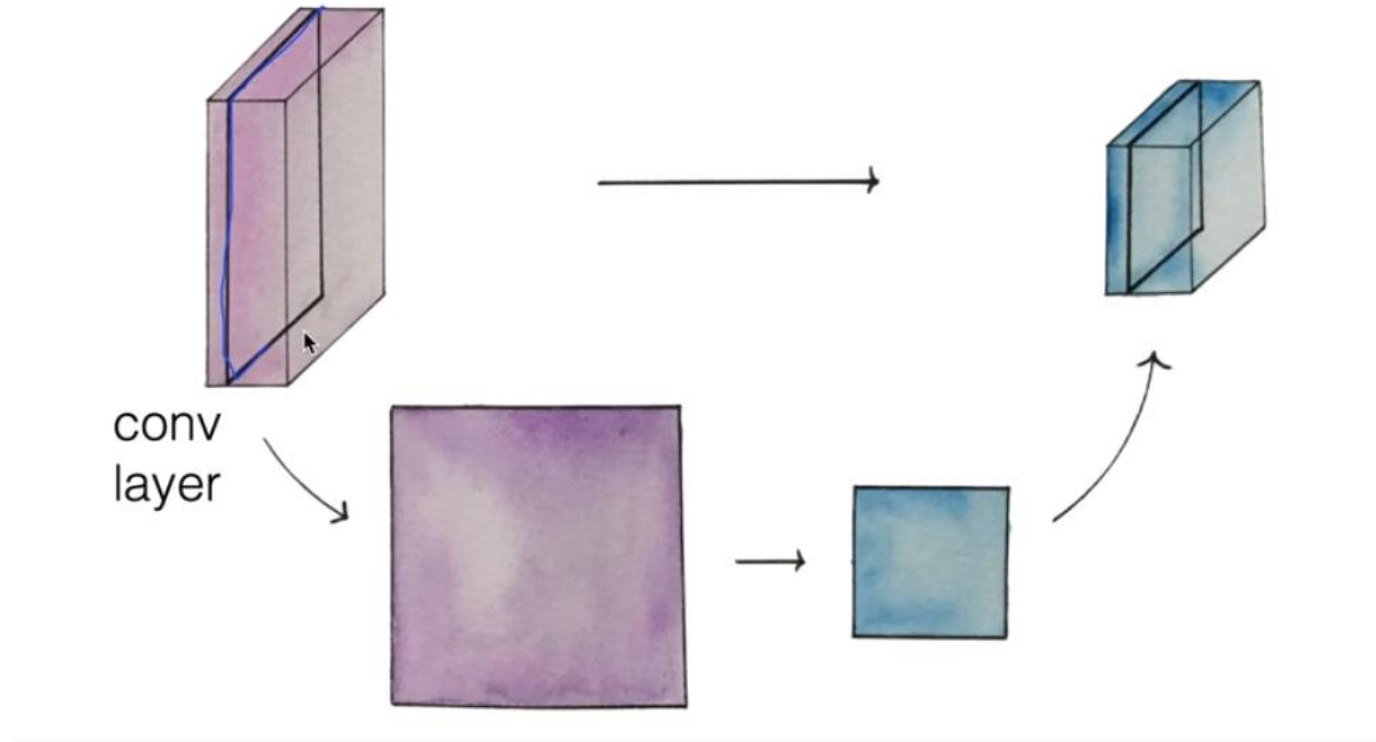
차원 감소 필요 -> 풀링의 역할

Pooling layer

- 공간 차원 축소
- 노이즈 감소
- Ex) 맥스 풀링 : 영역내 최댓값을 대푯값으로

Pooling layer

- Like 샘플링



Fully connected layer

- DNN과 동일
- 특징을 추출, 분류 수행

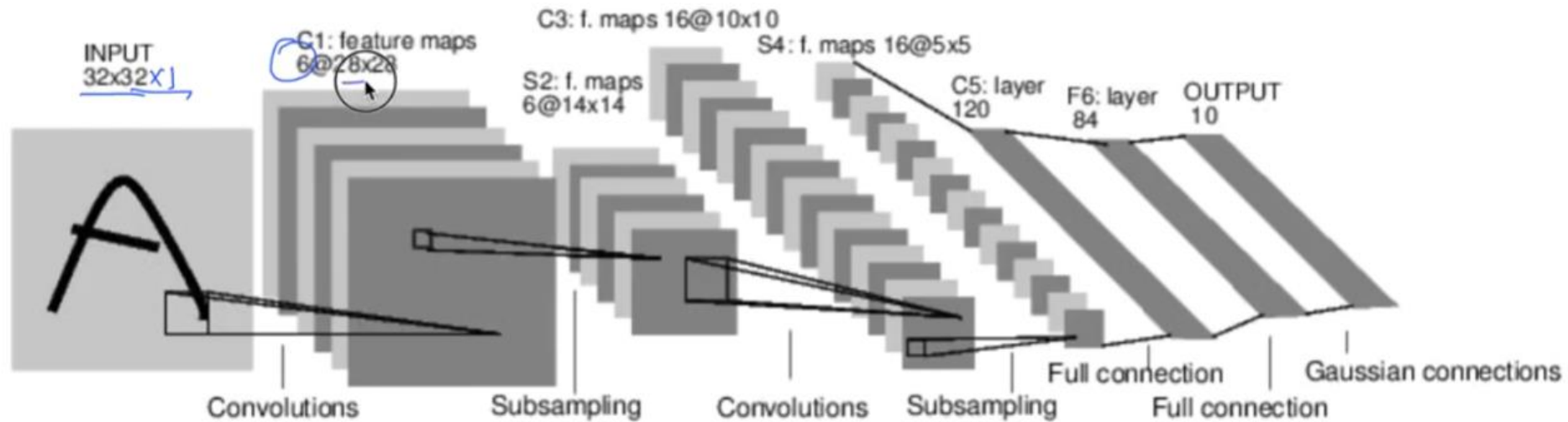
CNN 모델의 역사

- Lenet-5
- Alexnet
- Googlenet
- Resnet

각 모델의 특징, 키워드 위주로

Lenet-5

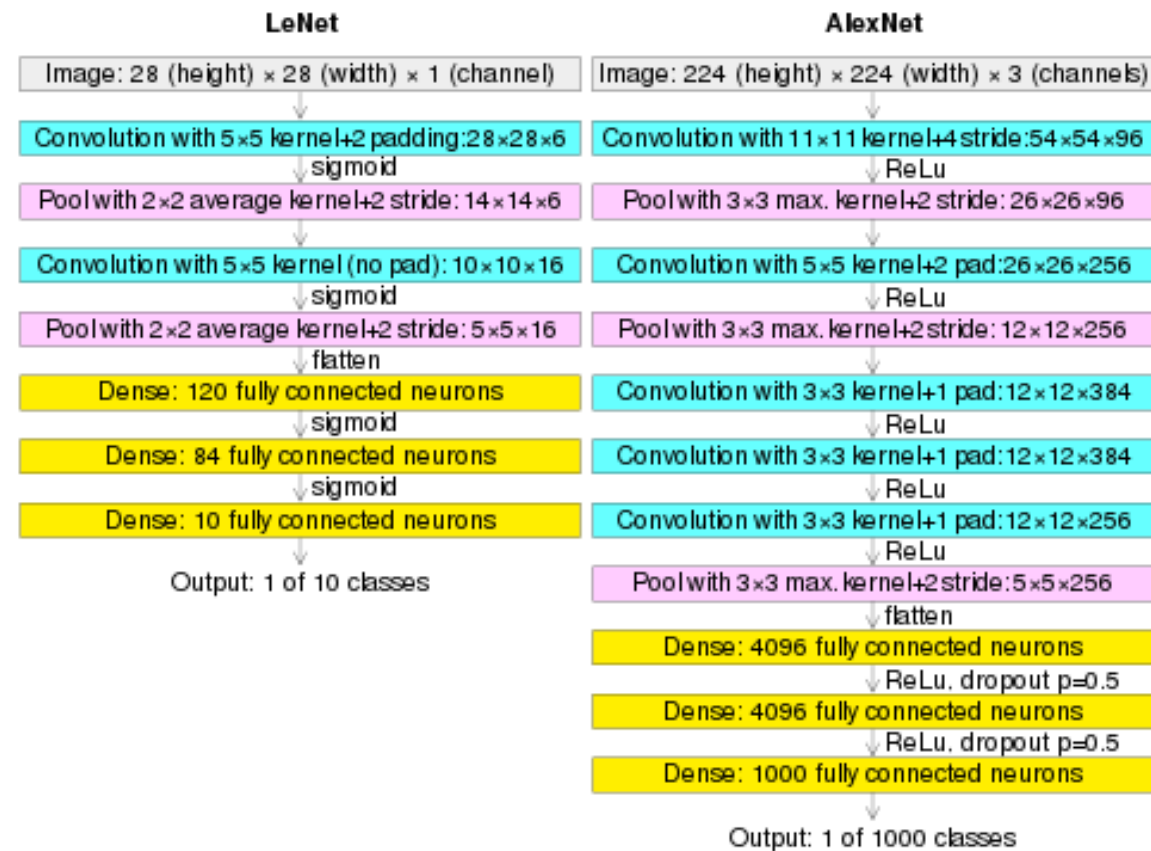
- CNN의 조상님



- 지금까지 배운 기본적 내용

Alexnet

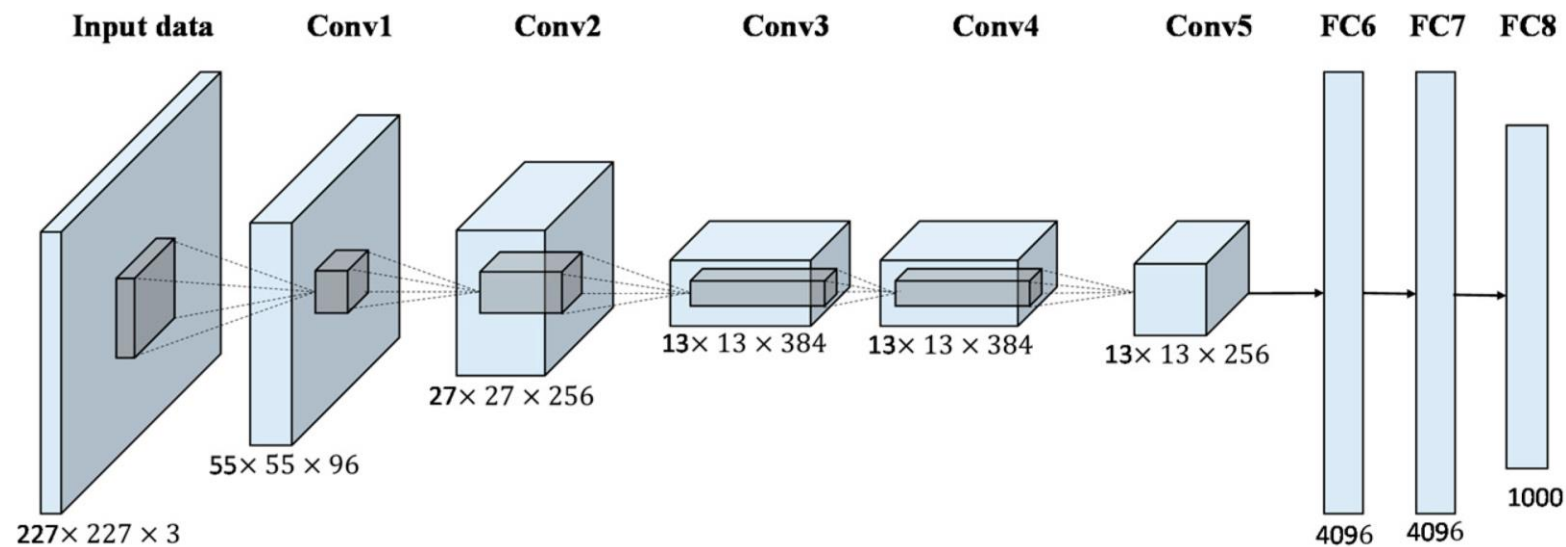
- 2012년 ImageNet 대회에서 우승
- 5개 Conv, 3개 Pool
- 3개 FC layer
- Sigmoid가 아닌 Relu를 activation으로 사용



Alexnet

- 대규모 딥러닝 모델(param多)

- GPU가속화



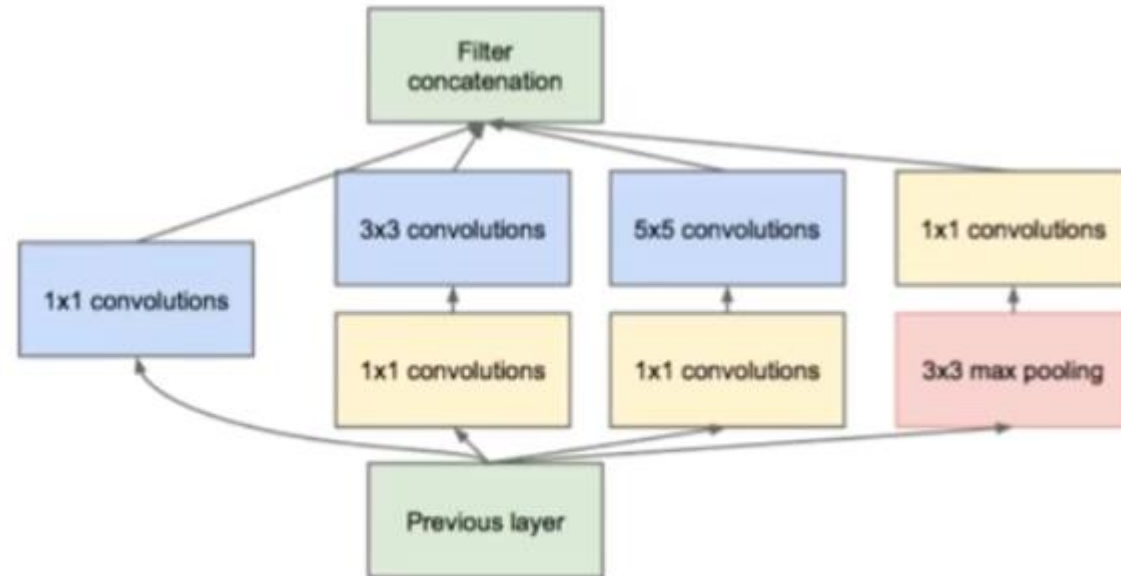
GPGPU

- CPU의 연산을 GPU가 같이 함
- 여러 개 Thread -> 병렬 처리↑ -> 연산 속도↑
- 대규모 데이터 처리



Googlenet

- Inception model
- 다중 크기 Conv filter
- *1x1 filter*
- Batch normalization



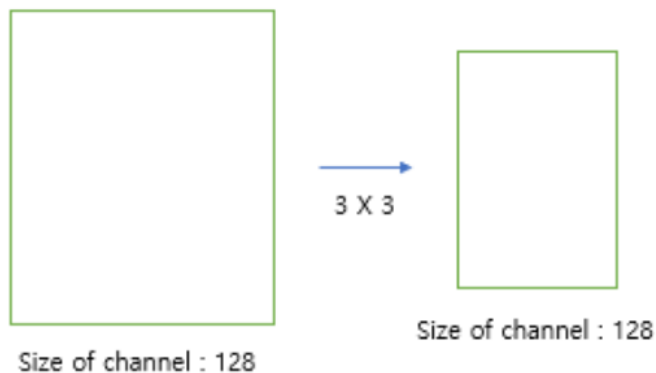
- 2014년 ILSVRC 우승

Inception block

1x1, 3x3 과 같은 다중 크기 conv filter 적용

Inception block

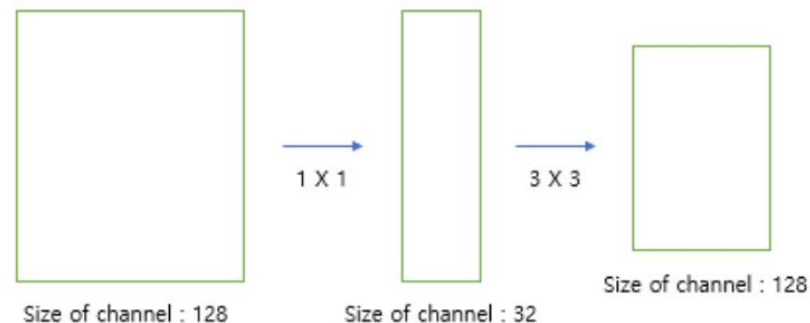
3x3필터로 한번



이 때 사용하게 될 파라미터의 수는 다음과 같다.

$$3 \times 3 \times 128 \times 128 = 147,456$$

1x1 한번, 3x3 한번

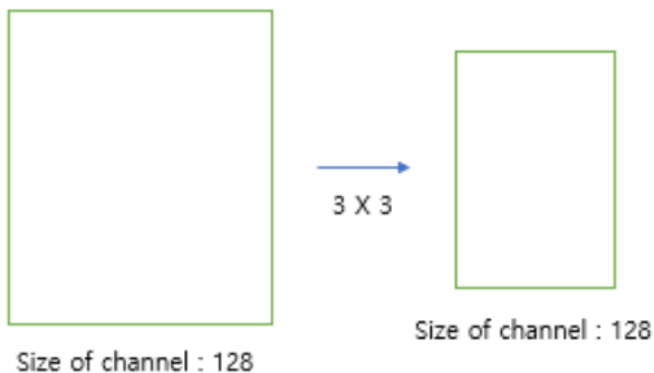


이 때 사용하게 될 파라미터의 수는 다음과 같다.

$$1 \times 1 \times 128 \times 32 + 3 \times 3 \times 32 \times 128 = 40,960$$

Inception block

3x3필터로 한번

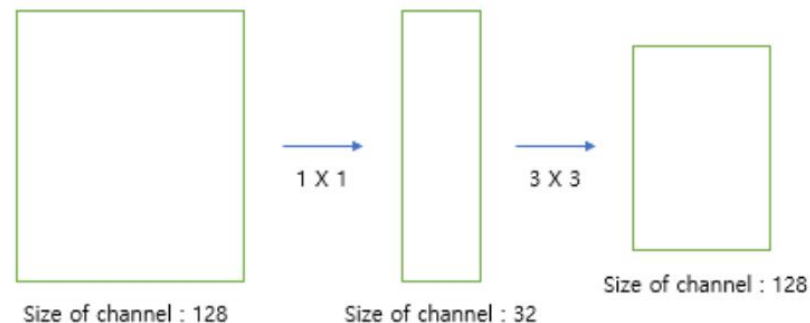


이 때 사용하게 될 파라미터의 수는 다음과 같다.

$$3 \times 3 \times 128 \times 128 = 147,456$$

Param 감소!

1x1 한번, 3x3 한번



이 때 사용하게 될 파라미터의 수는 다음과 같다.

$$1 \times 1 \times 128 \times 32 + 3 \times 3 \times 32 \times 128 = 40,960$$

Googlenet

- AlexNet (2012, 8 Layers) / 60M size of parameters
- VGGNet (2014, 19 Layers) / 110M size of parameters
- GoogLeNet (2014, 22 Layers) / 4M size of parameters

Resnet

- Residual Block
- 매우 깊은 layer 구조
- 전역 평균 풀링

ILSVRC 2015 winner (3.6% top 5 error)

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)



ResNet, 152 layers
(ILSVRC 2015)



Resnet

깊은 layer를 만들자

모델 training 성능 ↑

gradient vanishing 문제 => layer 사이 shortcut

Residual Block

- 입력 데이터를 건너뛰는 (skip) 연결

Grad 소실 문제 완화

Googlenet

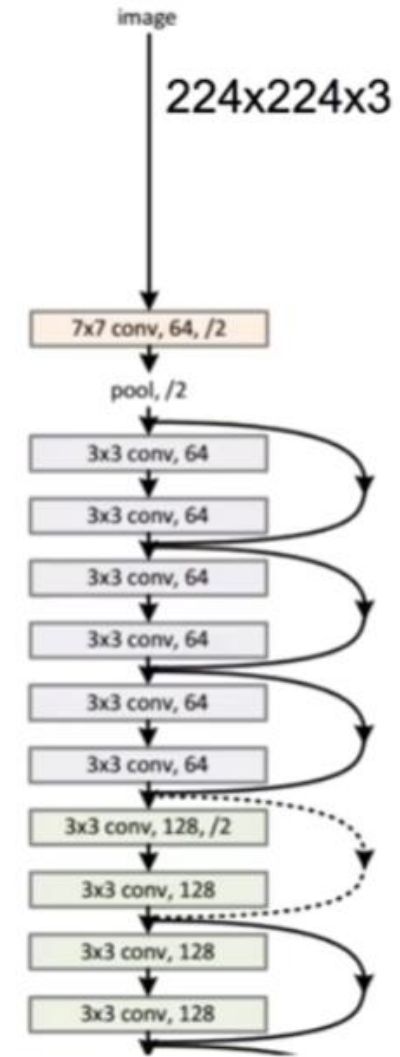
병렬 필터

Resnet

스킵(잔차) 연결

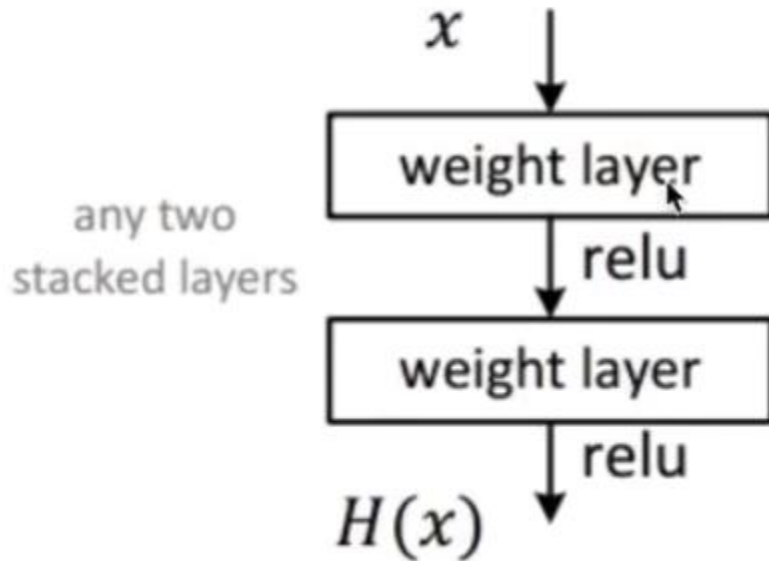
Residual Block

34-layer residual

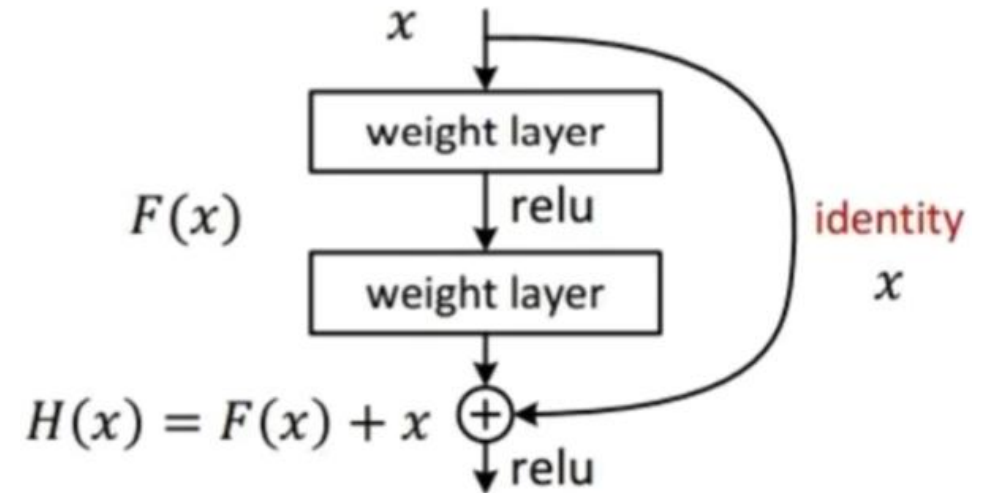


Residual Block

- Plain net



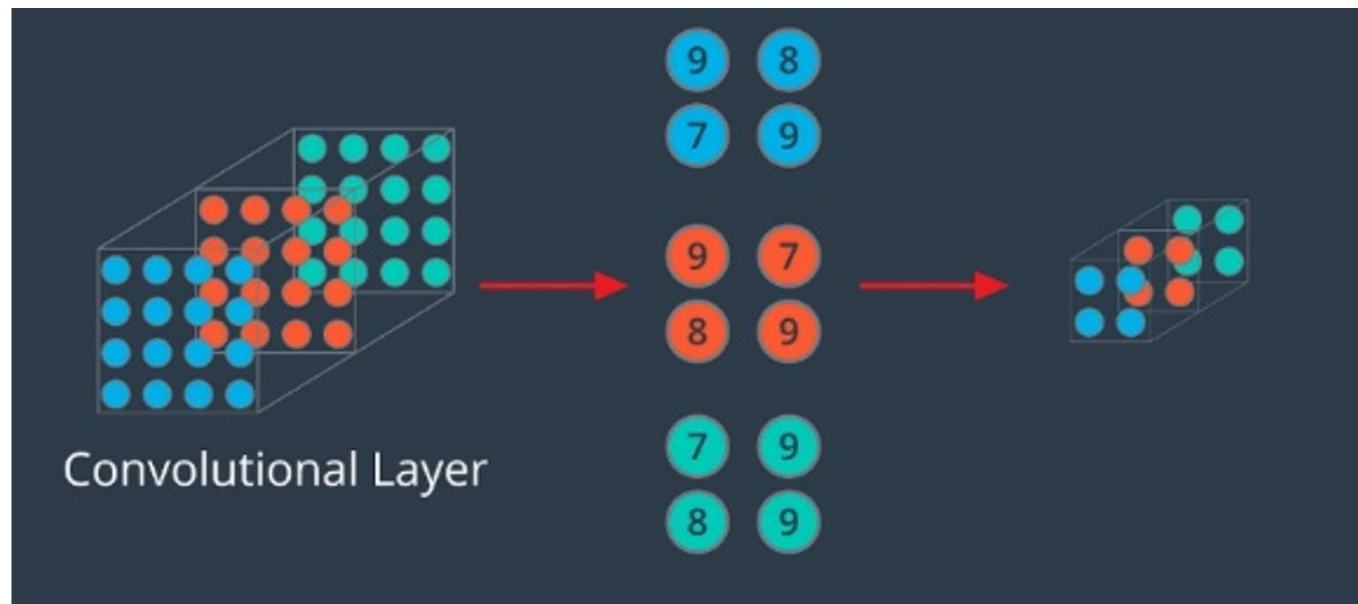
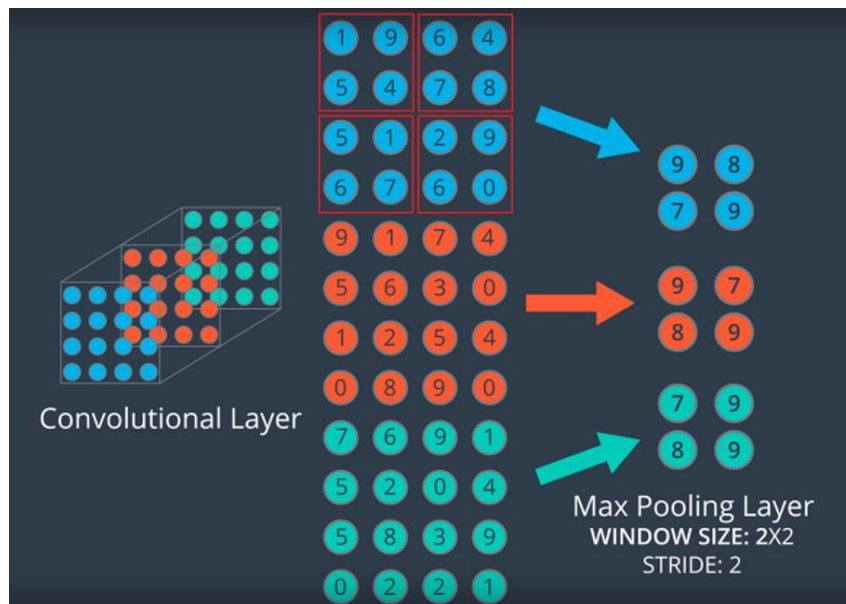
- Residual net



전역 평균 풀링

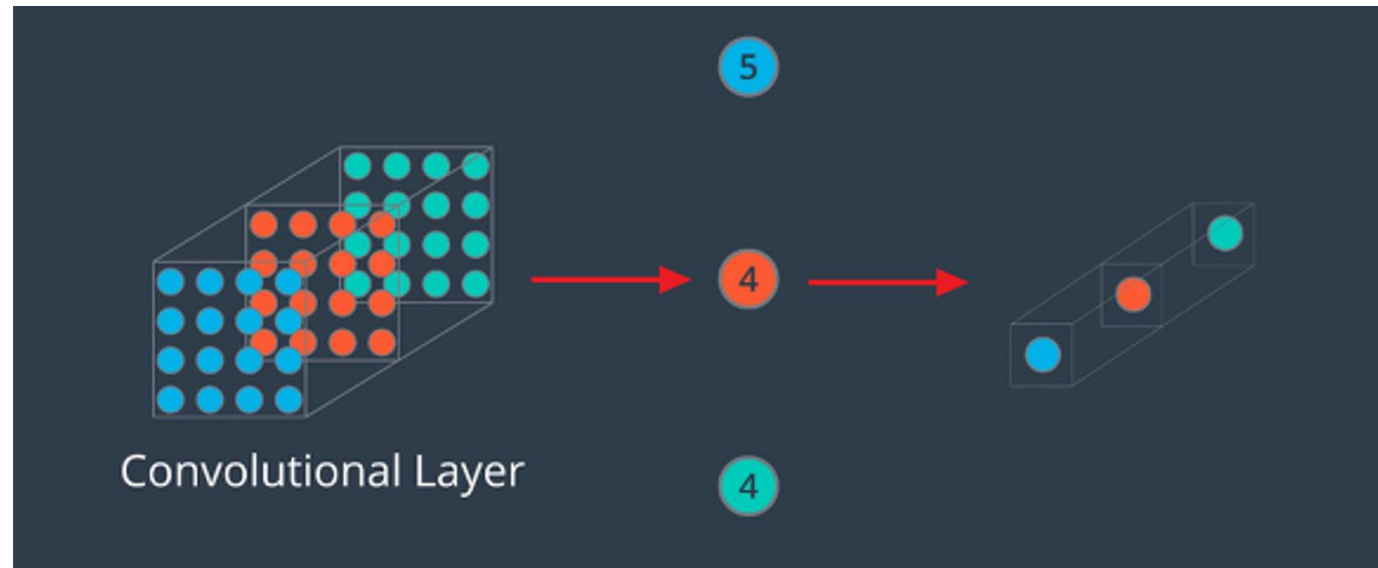
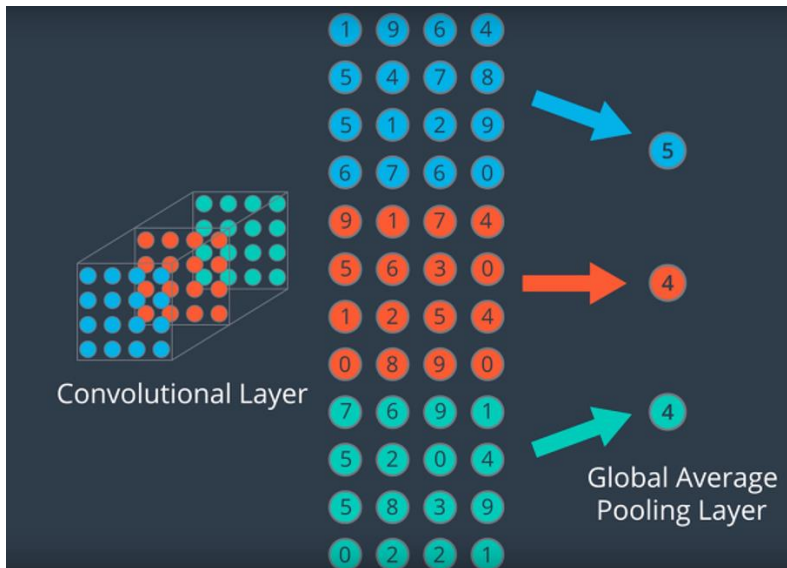
- Global Average Pooling(GAP)
- FC layer를 없애기 위함(깊이 감소)

GAP VS MP



GAP VS MP

- 급격히 feature를 줄임 (전역의 평균)
- 1차원 벡터로



그 외 CNN

- CNN for sentence classification
- ALphaGO



발표를 들어주셔서 감사합니다 😊

프로젝트 제출하기



1

프로젝트 등록

2

제출 완료

Zip

STEP 1 제출하고자 하는 프로젝트를 하나의 압축파일로 만들어서 등록하세요.

파일등록

STEP 2 리뷰어에게 강조하거나 전할 내용이 있으면 작성하세요. (선택)

내용을 입력하세요.

STEP 3 약속합니다!

1. 나는 다른 학습자의 제출내용을 표절하지 않았음을 약속합니다.
2. 나는 활용한 내용의 출처를 표시하였습니다. (웹사이트, 책, 포럼, 블로그, Github 등)
3. 나는 표절 검사를 하는 것에 동의하고, 위 사항을 위반 시 수강 취소될 수 있음을 숙지하였습니다.

☐ 동의합니다.