



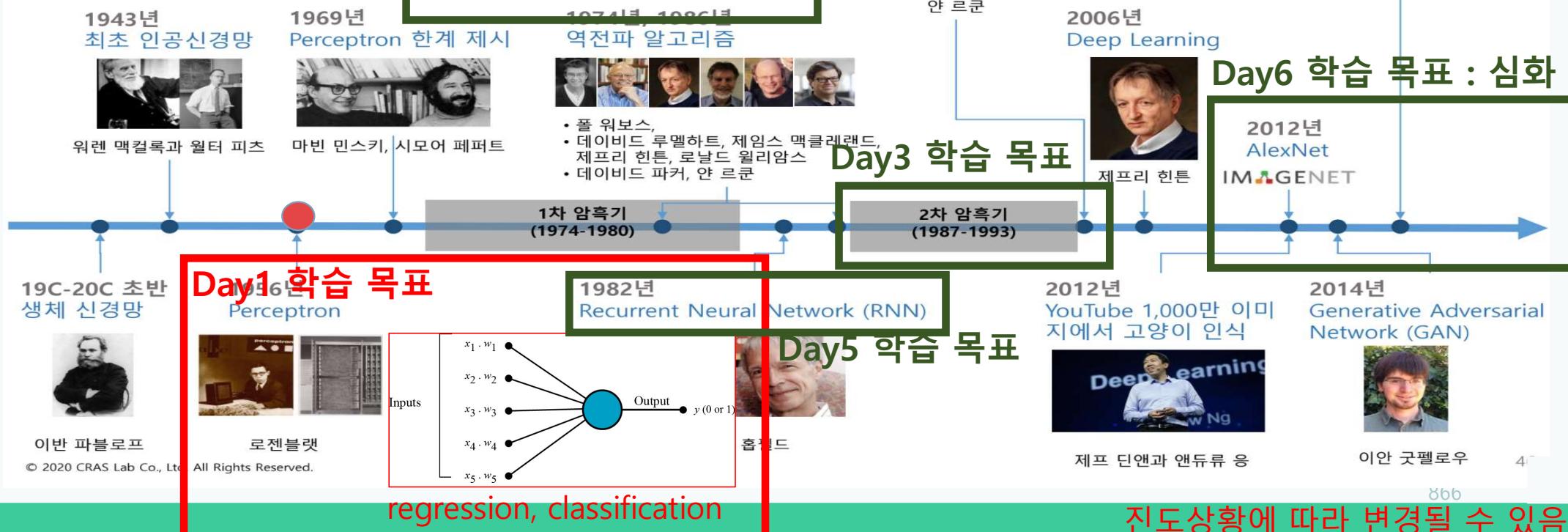
DAY 5

CNN 이론 실습
RNN/LSTM/GRU 이론 실습

0. 복습

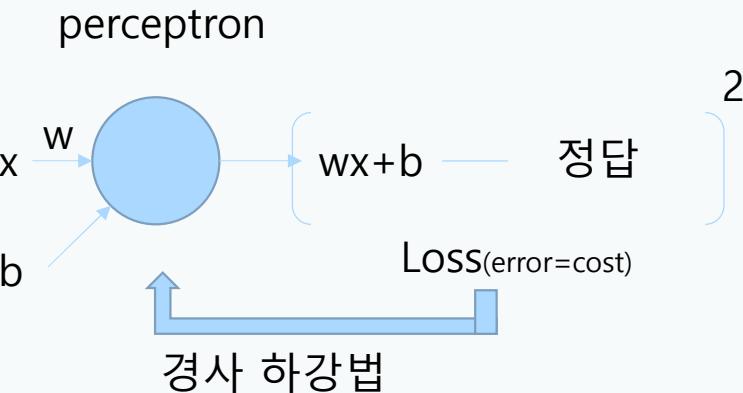
Deep Learning

신경망의 70년



0. 복습

1. Perceptron

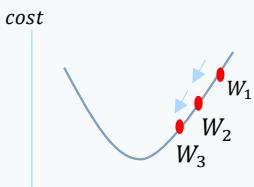


$$\begin{aligned} & (WX - Y)^2 \\ & = W^2X^2 - 2WXY + Y^2 \\ & = AW^2 - BW + C \\ & = \text{cost(error)} \end{aligned}$$

기울기

$$W_{update} = W - \alpha \frac{\partial}{\partial W} \text{cost}(W)$$

러닝 레이트



Regression

출력 함수 = None

loss = mse

Binary classification

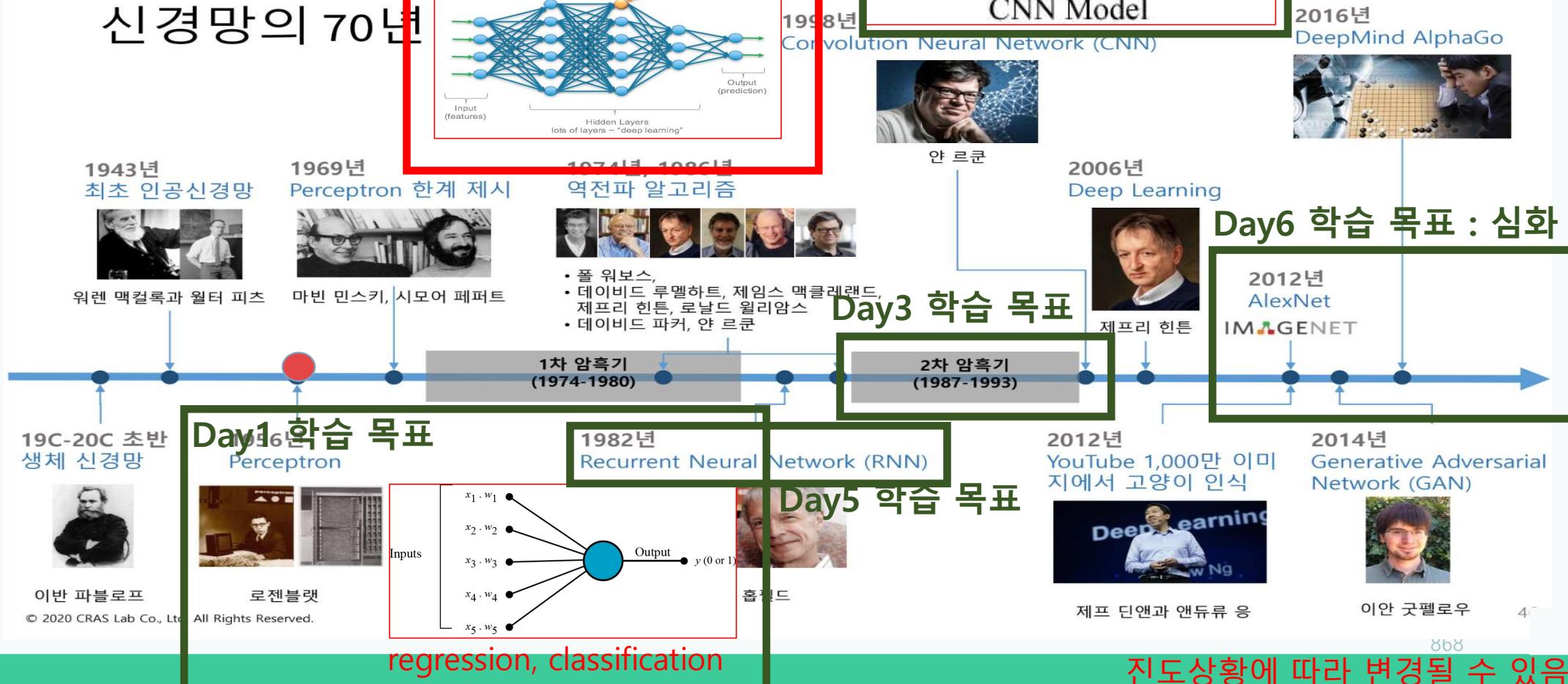
출력함수 = sigmoid

loss = binary crossentropy

0. 복습

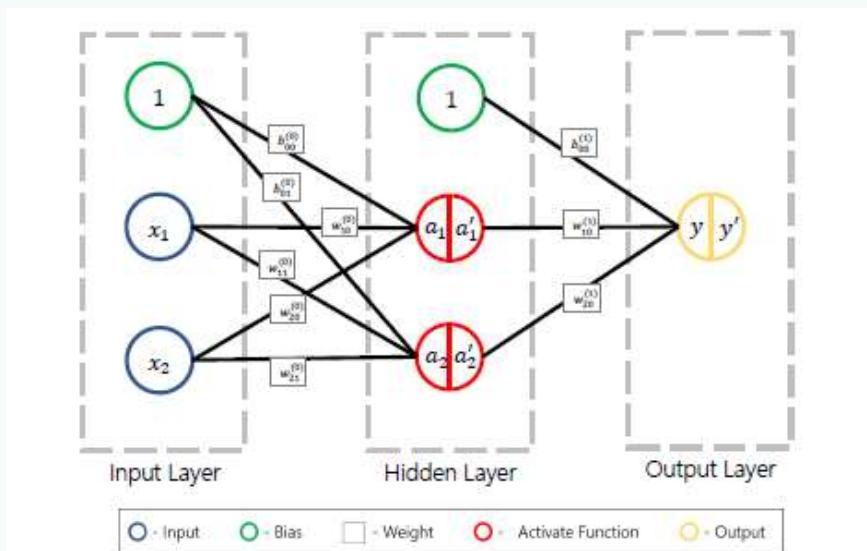
Deep Learning

신경망의 70년



0. 복습

2. Multi Layer Perceptron



다층 구조가 가능 하기 위해

activation function 사용 = sigmoid

hidden layer의 역할

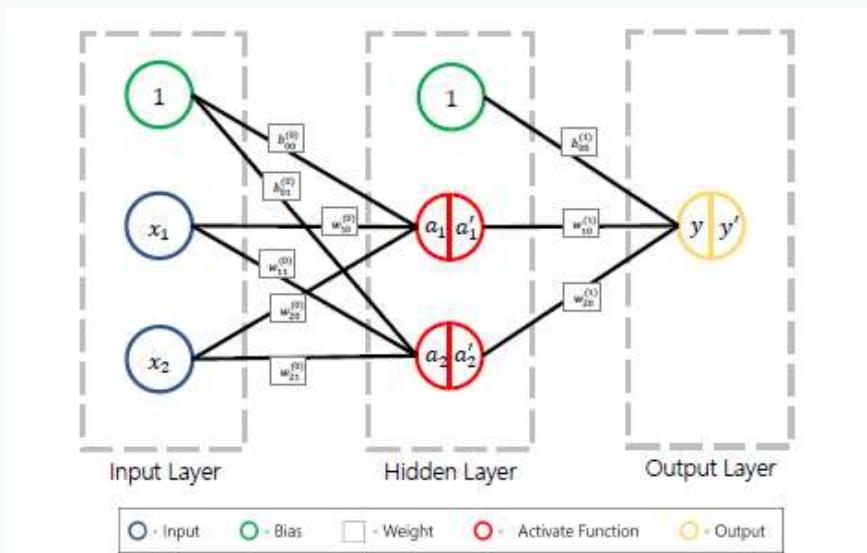
output layer가 잘 동작하도록 입력 데이터를
representation 해주는 역할

입력 차원 : (batch size, 데이터 변수)

모델 학습 알고리즘 : backpropagation

0. 복습

2. Multi Layer Perceptron



Regression

출력 함수 = None

loss = mse

Binary classification

출력함수 = sigmoid

loss = binary crossentropy

Categorical classification

출력함수 = softmax

loss = categorical crossentropy

(one hot encoding 사용)

0. 복습

2. Multi Layer Perceptron

다층 구조가 가능 하기 위해

여기서 hidden layer의 activation function과 output layer의 activation function은 다르다!

Hidden layer의 activation function

1. 여러 layer를 쌓는 효과를 보기위해 비선형적이어야함.
2. Layer들의 w,b가 잘 업데이트 되기 위해 오차 역전파에서 오차의 미분값이 잘 전달되도록해야함
3. 미분 가능해야함

Output layer의 activation function은

Task에 따라 선정.

Regression 의 activation function은 None (regression은 출력값이 숫자값으로 입력*w + b값이 예측값)

Classification의 activation function은 sigmoid or softmax (classification은 0 or 1의 제한된 출력)

0. 복습

2. Multi Layer Perceptron

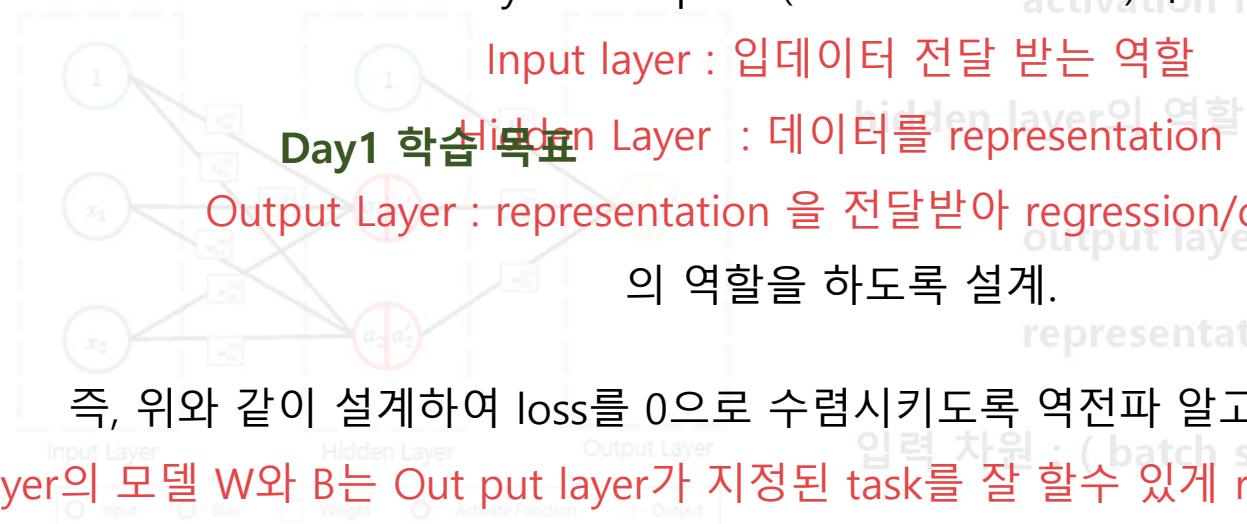
다층 구조가 가능 하기 위해

Multi Layer Perceptron(=Neural Network) 구조 할때

Input layer : 입데이터 전달 받는 역할

Hidden Layer : 데이터를 representation

Output Layer : representation 을 전달받아 regression/classification
의 역할을 하도록 설계.



즉, 위와 같이 설계하여 loss를 0으로 수렴시키도록 역전파 알고리즘으로 학습하면

Hidden Layer의 모델 W와 B는 Out put layer가 지정된 task를 잘 할수 있게 representation되도록 학습되어 짐

Output Layer의 w와 b는 지정된 task를 잘하도록 학습 되어 짐.

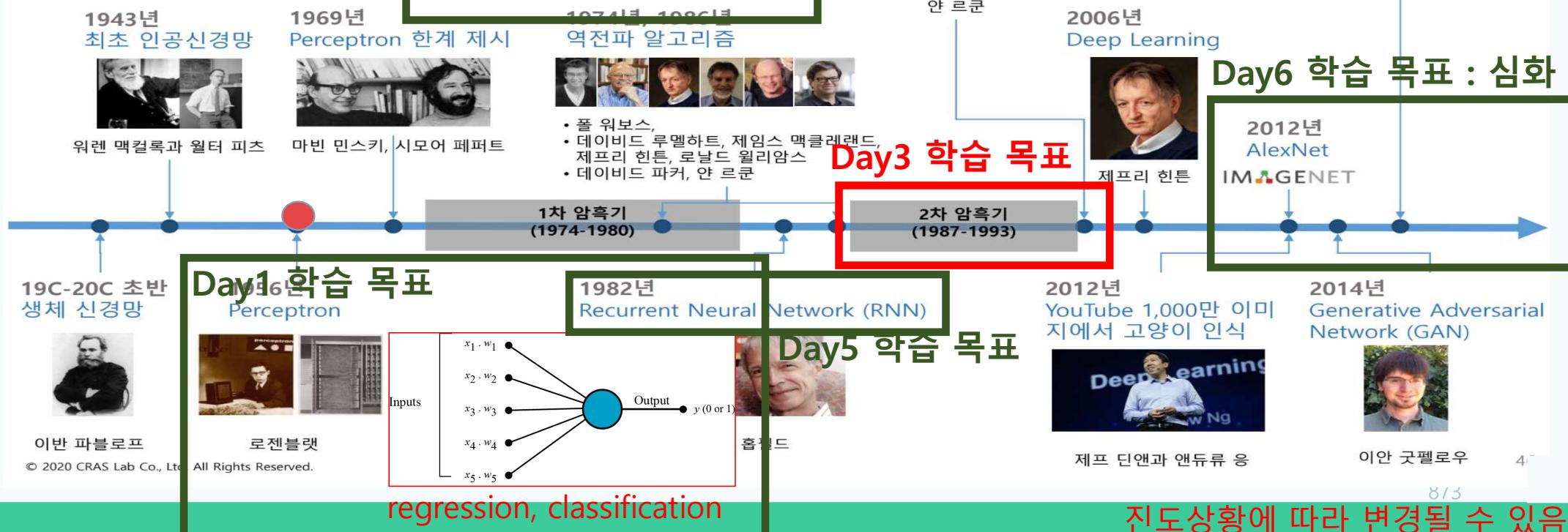
즉, 결론적으로 학습이 잘되면 각 layer들은 각각의 역할을 잘 수행한다고 해석함.

학습이 잘 안되면 위의 역할을 잘 수행 못한다고 해석함.

0. 복습

Deep Learning

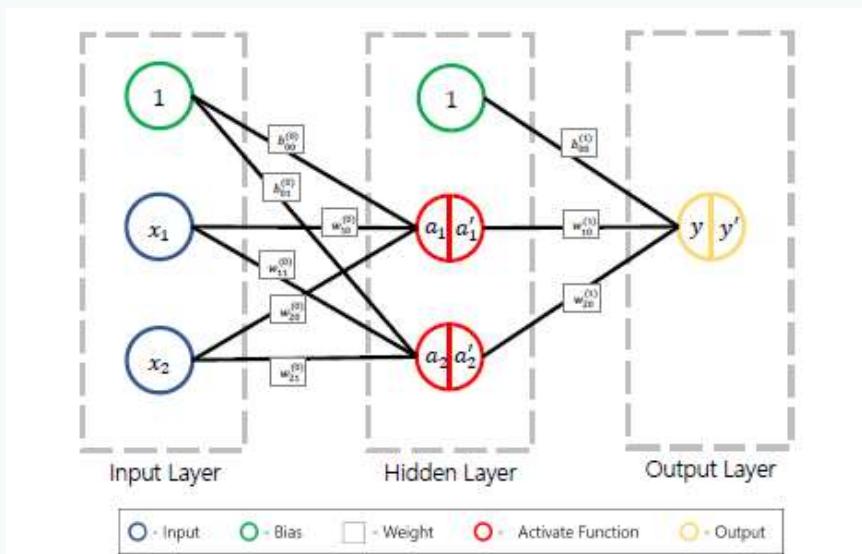
신경망의 70년



0. 복습

2. Multi Layer Perceptron

Multi Layer Perceptron 문제



1. 오비피팅

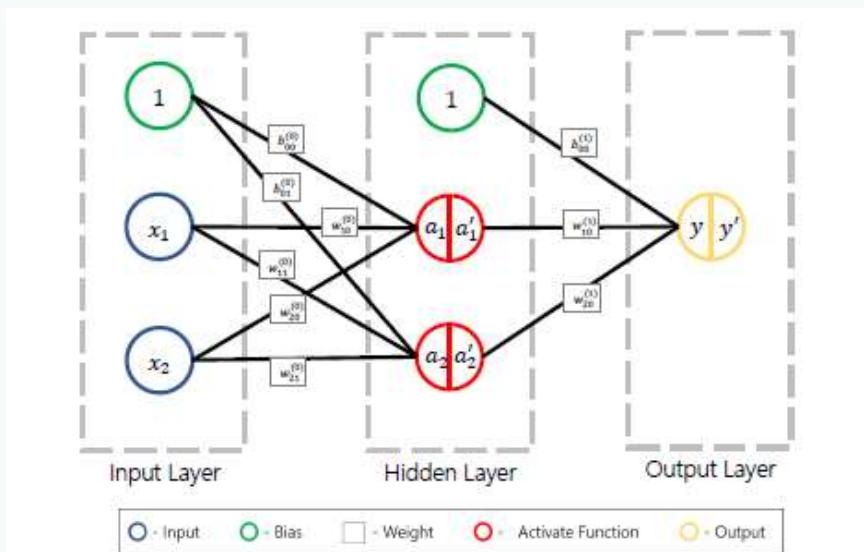
2. 느린 학습 속도

3. local minima

4. vanishing gradient

0. 복습

2. Multi Layer Perceptron



Multi Layer Perceptron 대표적 해결법

1. batch normalization

2. dropout

3. regularization

4. small model

5. data augmentation

6. 새로운 activation 함수

7. 좋은 optimizer

0. 복습

3. 학습에서 중요한 것

데이터를 모을 때 현장과 비슷하고 발생할 모든 케이스의 데이터를 많이 모아야함.

학습/검증/시험 데이터 나눠서 학습시 성능 확인을 하고 오버피팅 확인을 해야함

데이터를 나눌 때 분포가 고르게 되어야함

학습 모델 웨이트가 고루 잘 퍼져잇는게 일반적으로 잘 학습되었다고 함.

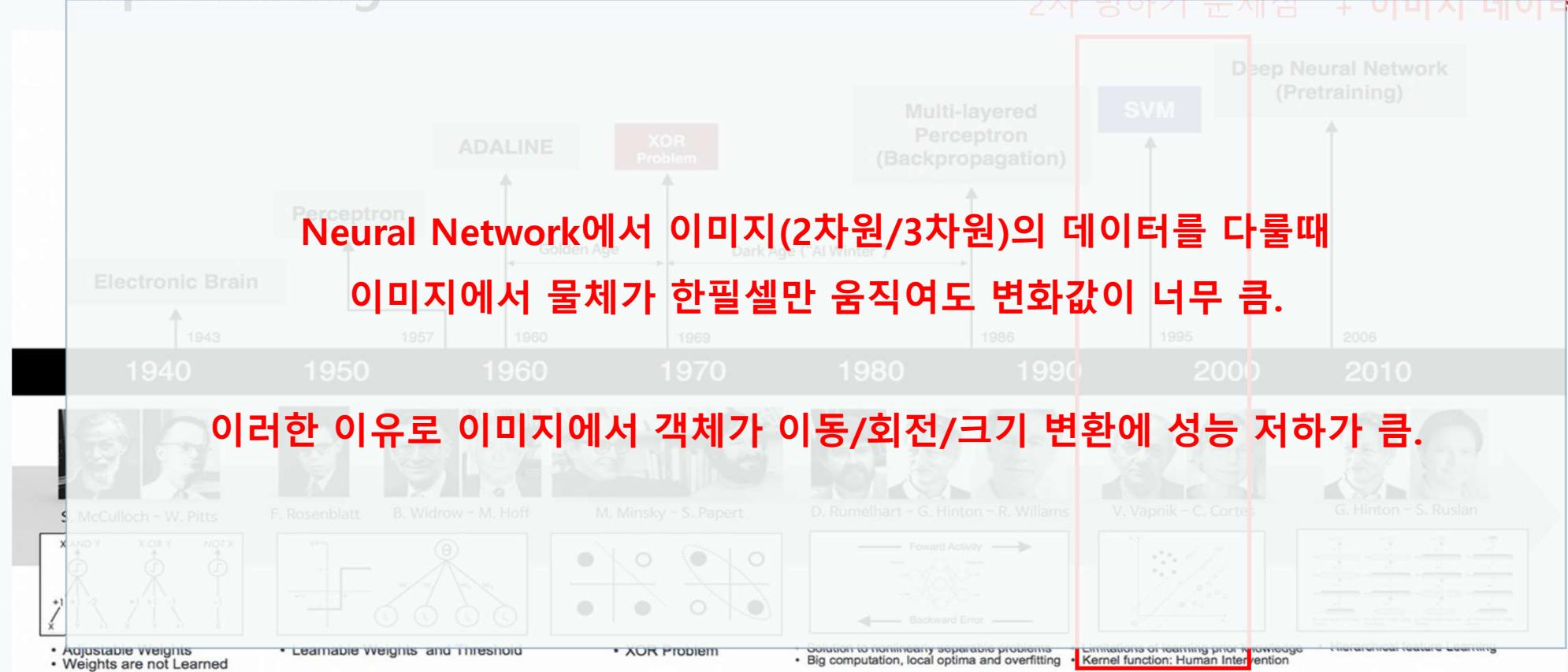
러닝 레이트, 모델 파라미터 초기값, 모델 설계, batch size, epoch 등이 중요

검증데이터는 학습시 모델 업데이트에 적용되지 않음

0. 복습

Deep Learning

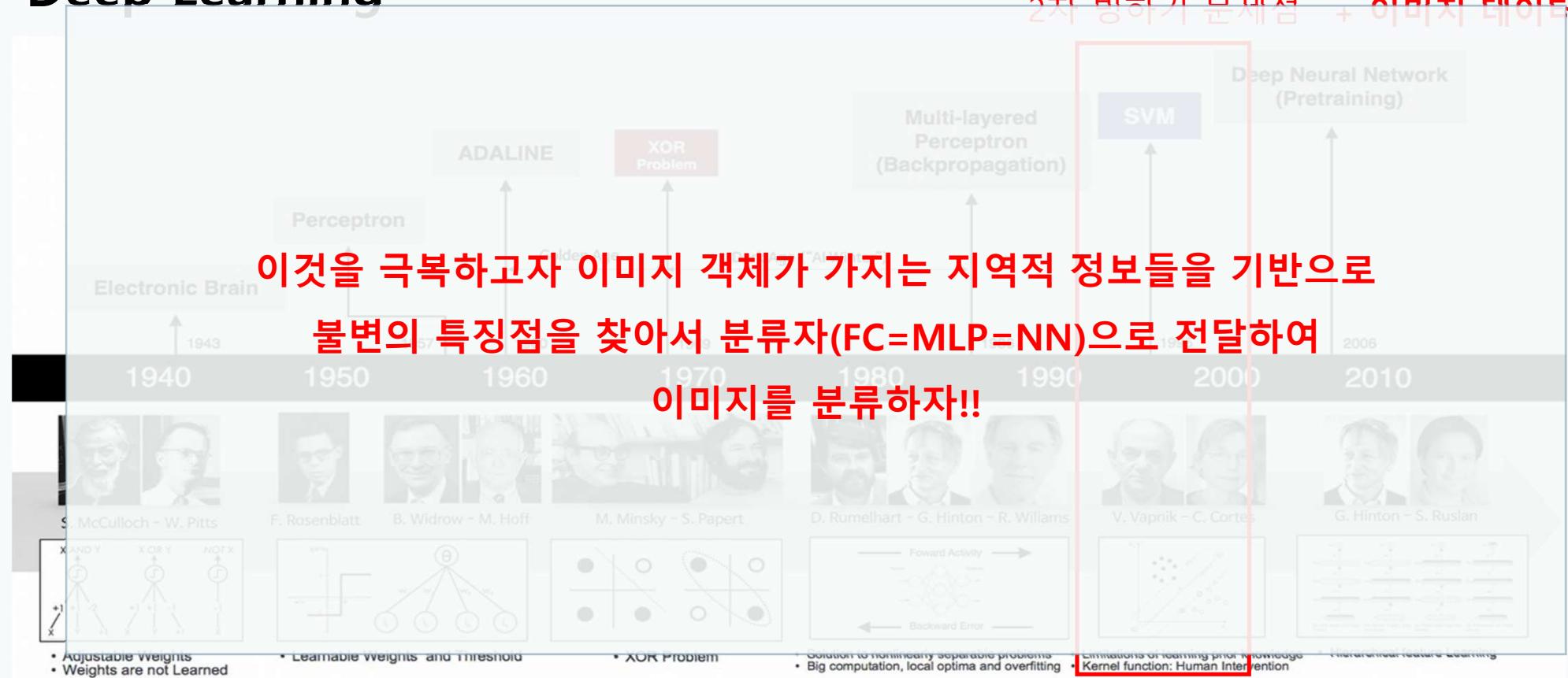
Neural Network 의 + 이미지 다루기
2차 방하기 문제점 + 이미지 데이터 문제



0. 복습

Deep Learning

Neural Network 의 + 이미지 다루기
2차 방하기 문제점 + 이미지 데이터 문제



0. 복습

Deep Learning

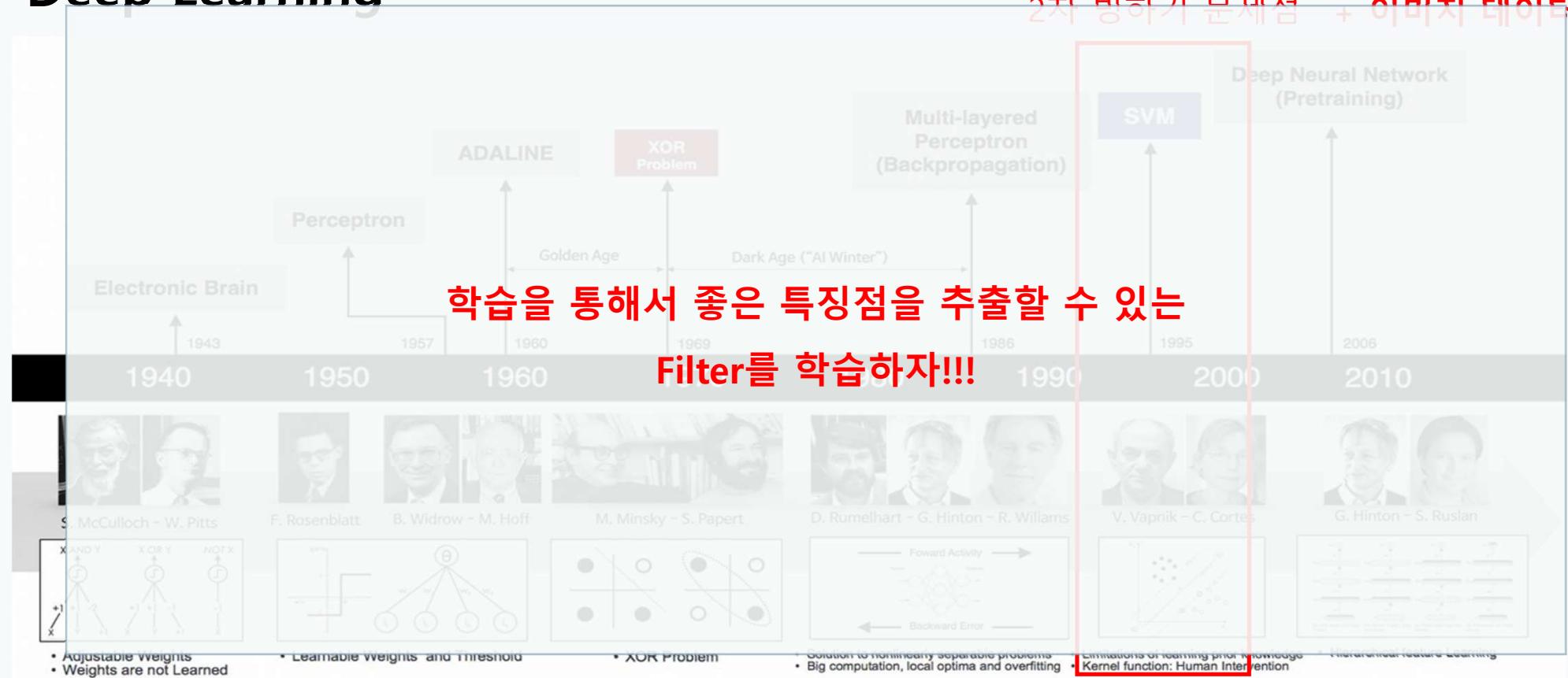
Neural Network 의 + 이미지 다루기
2차 방하기 문제점 + 이미지 데이터 문제



0. 복습

Deep Learning

Neural Network 의 + 이미지 다루기
2차 방하기 문제점 + 이미지 데이터 문제



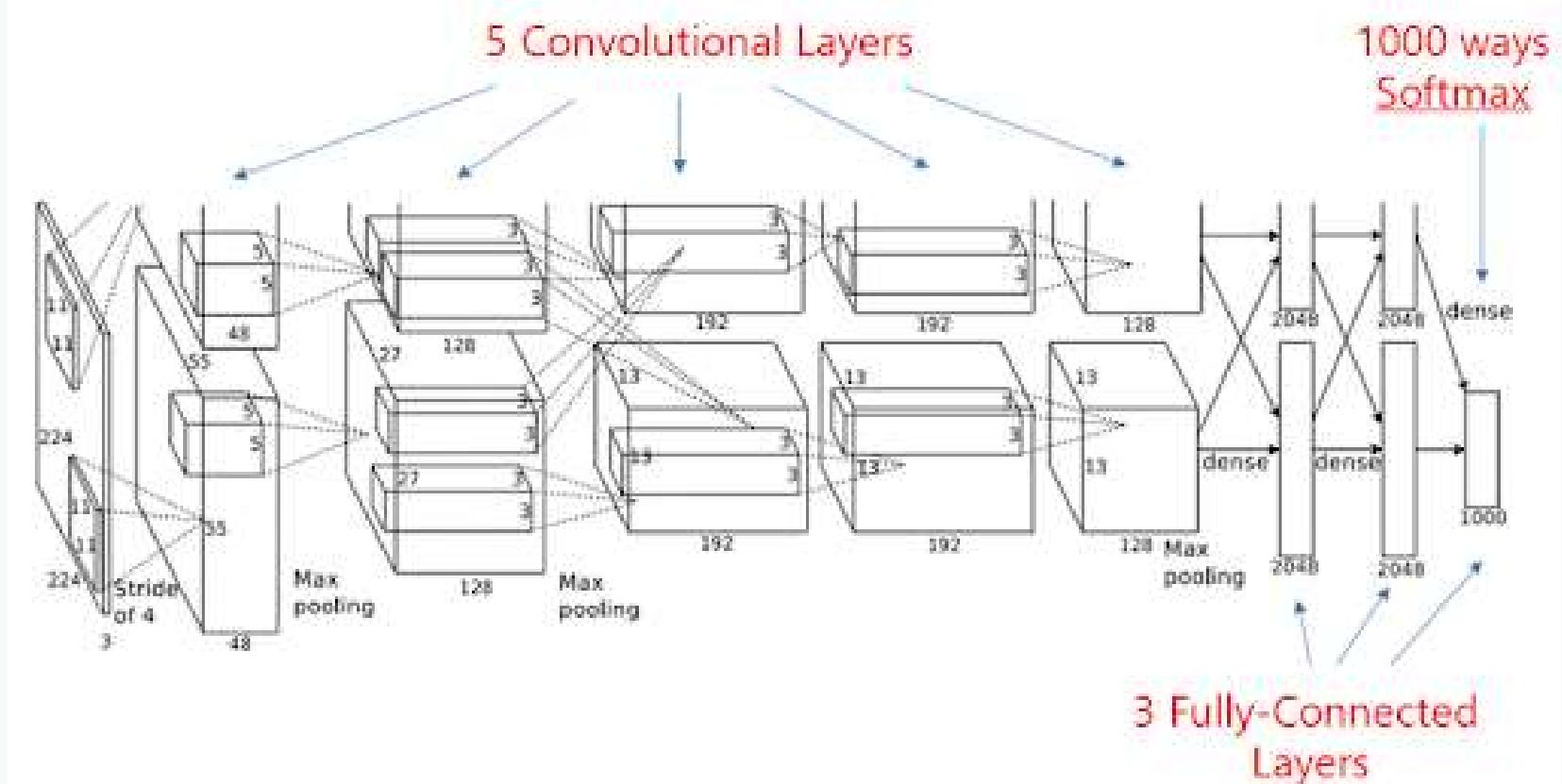
0. 복습

Convolutional
Neural Network

Convolution Layer(conv)

Pooling Layer(pool)

Fully Connected Layer(FC)



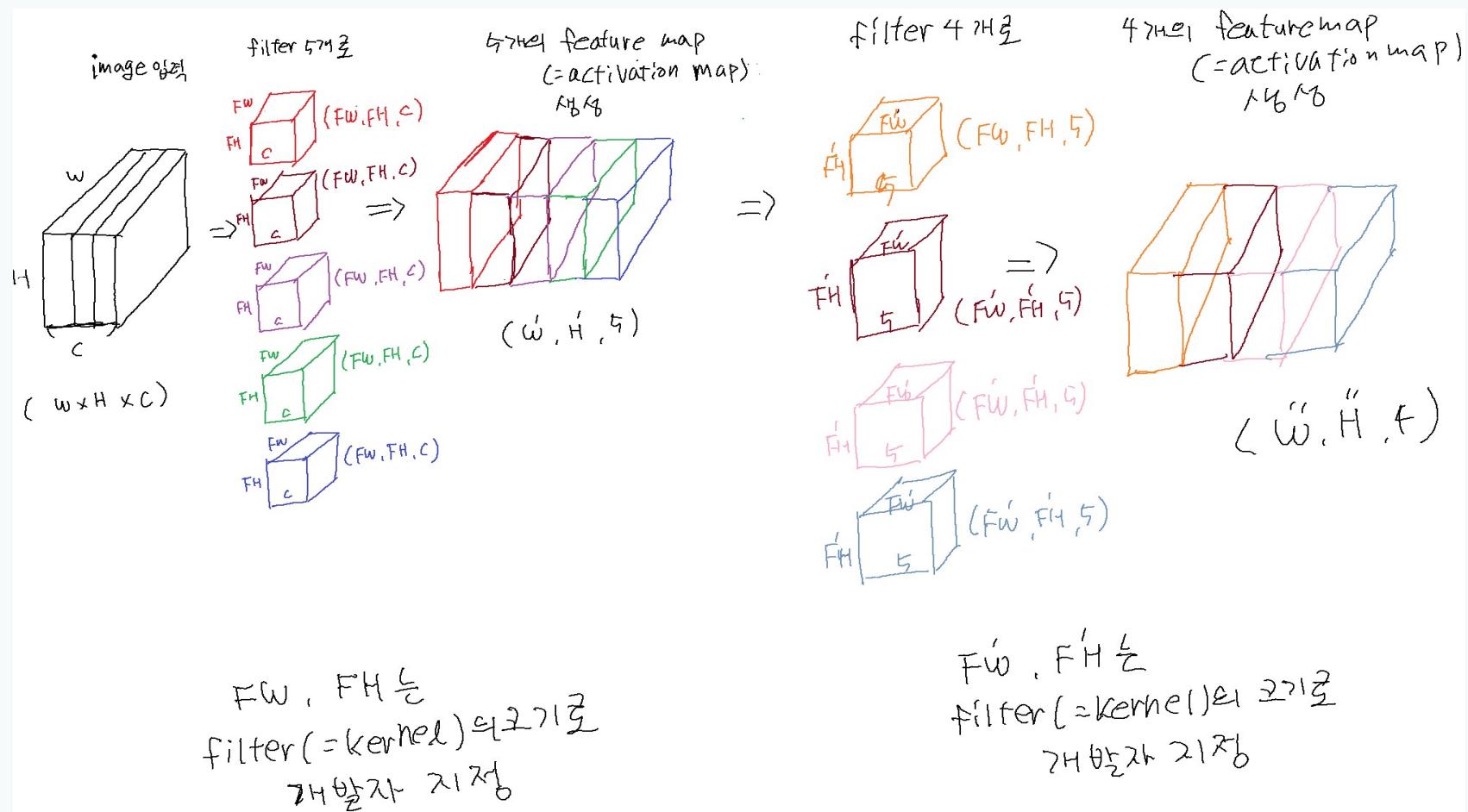
0. 복습

Convolutional Neural Network

Convolution Layer(conv)

Pooling Layer(pool)

Fully Connected Layer(FC)



0. 복습

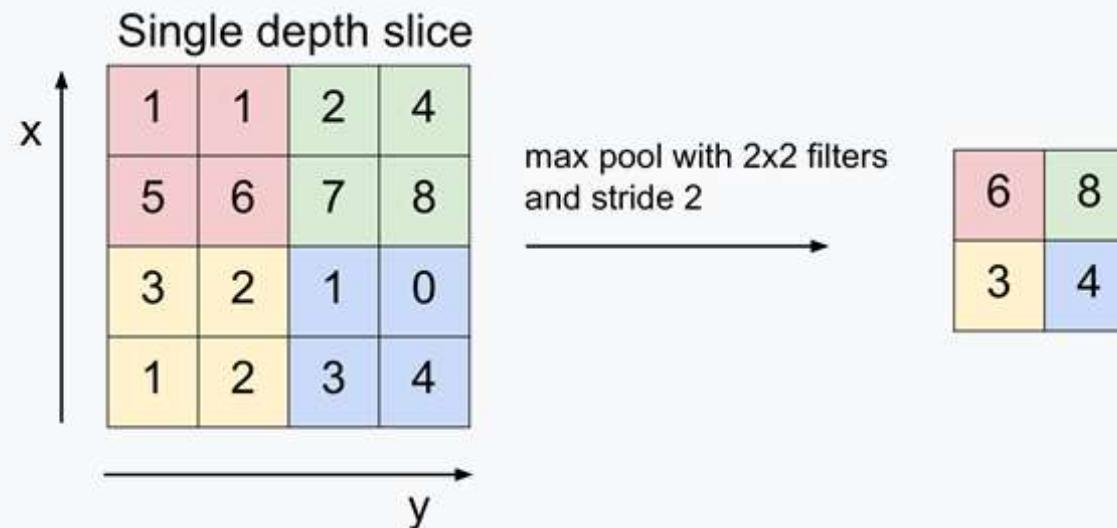
Convolutional
Neural Network

Convolution Layer(conv)

Pooling Layer(pool)

Fully Connected Layer(FC)

MAX POOLING



즉 filter(=kernel) 을 통해 feature을 만들었다면
만들어진 feature map에서 큰 특징을 가지는 값만 모아서
새로운 feature map을 만드는 과정

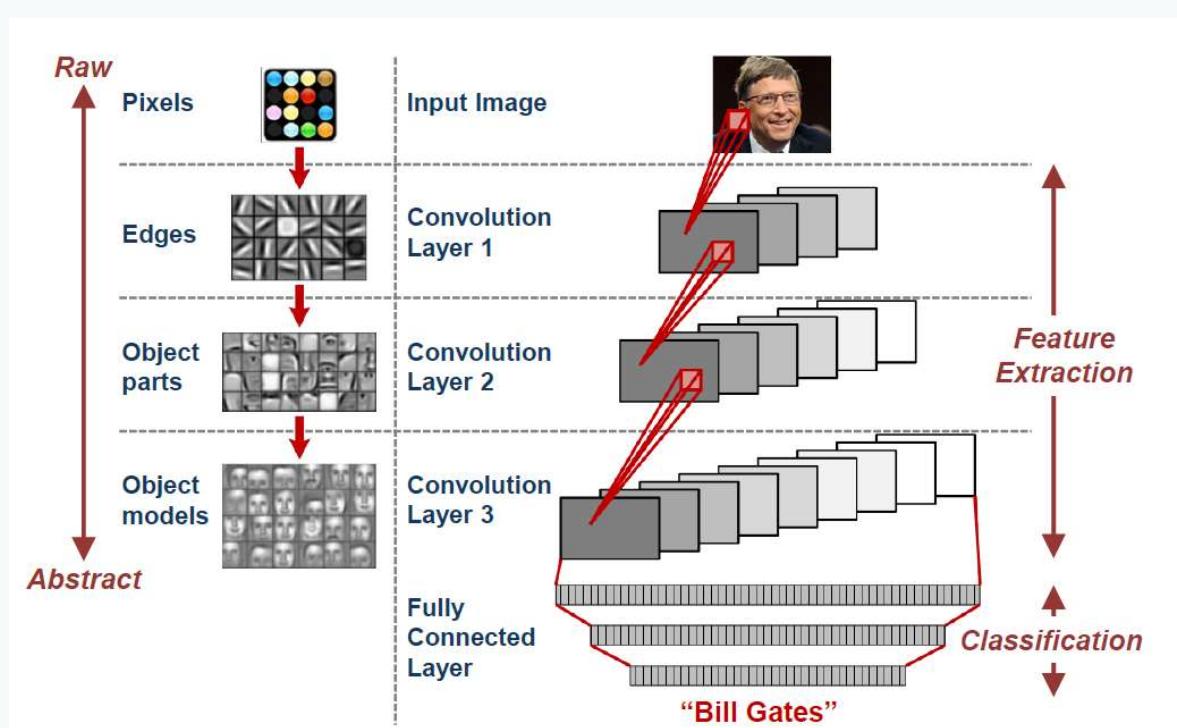
0. 복습

Convolutional
Neural Network

Convolution Layer(conv)

Pooling Layer(pool)

Fully Connected Layer(FC)



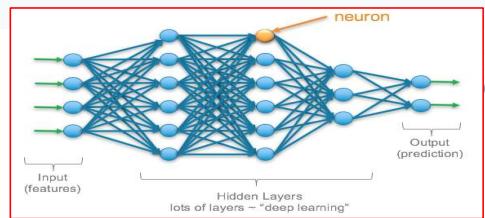
0. 오늘의 학습 목표

Deep Learning

신경망의 70년

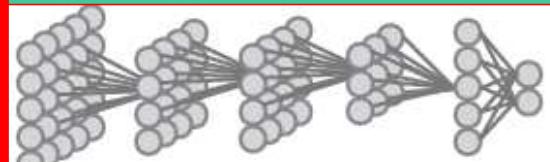
Day2 학습 목표

regression, classification



오늘의 학습 목표

Image classification, paper review



CNN Model



ян 르쿤

2016년
DeepMind AlphaGo



1943년
최초 인공신경망



워렌 맥컬록과 월터 피츠

1969년
Perceptron 한계 제시



마빈 민스키, 시모어 페퍼트

1974년, 1986년
역전파 알고리즘



- 폴 워보스,
- 데이비드 루멜하트, 제임스 맥클레란드,
- 제프리 힌튼, 로널드 윌리암스
- 데이비드 파커, 얀 르쿤

19C-20C 초반
생체 신경망



이반 파블로프

Day1 학습 목표

Perceptron

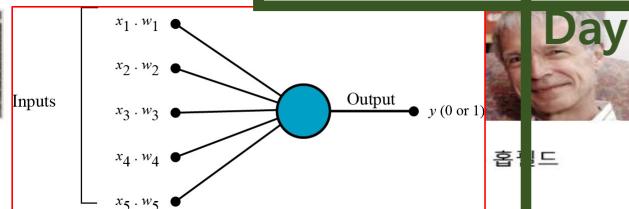


로젠블랫

© 2020 CRAS Lab Co., Ltd

All Rights Reserved.

1982년
Recurrent Neural
Network (RNN)



Day3 학습 목표

1차 암흑기
(1974-1980)

2006년
Deep Learning



Day6 학습 목표 : 심화

2012년
AlexNet

IMAGENET

2012년
YouTube 1,000만 이미지에서 고양이 인식



제프 딘앤과 앤드류 응

2014년
Generative Adversarial
Network (GAN)



이안 굿펠로우

47

88b

진도상황에 따라 변경될 수 있음

0. 오늘 하루 동안 무엇을 하나요?

1. CNN Backbone 모델 – paper review

- Alex Net
- VGG Net
- Google Net
- ResNet

2. Transfer Learning

- Fine turn
- Feature Extraction

3. RNN

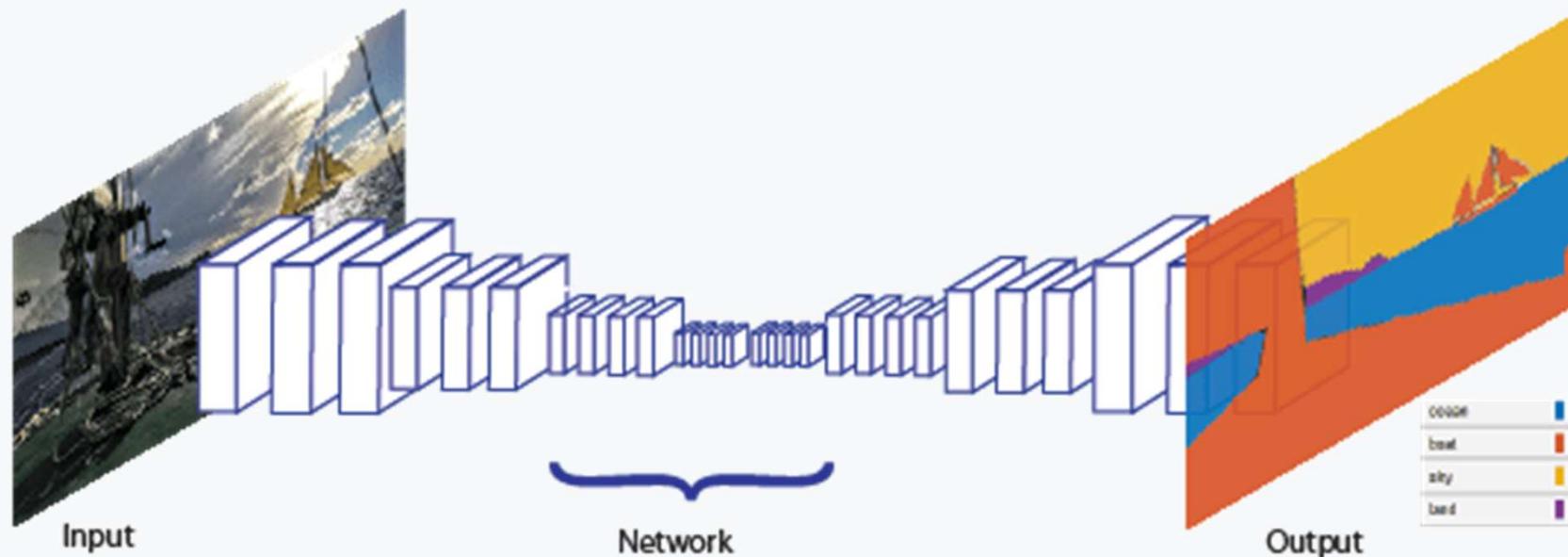
- RNN
- LSTM
- GRU

4. RNN 실습

- RNN 주식 예측하기
- LSTM 주식 예측 하기
- GRU 주식 예측하기

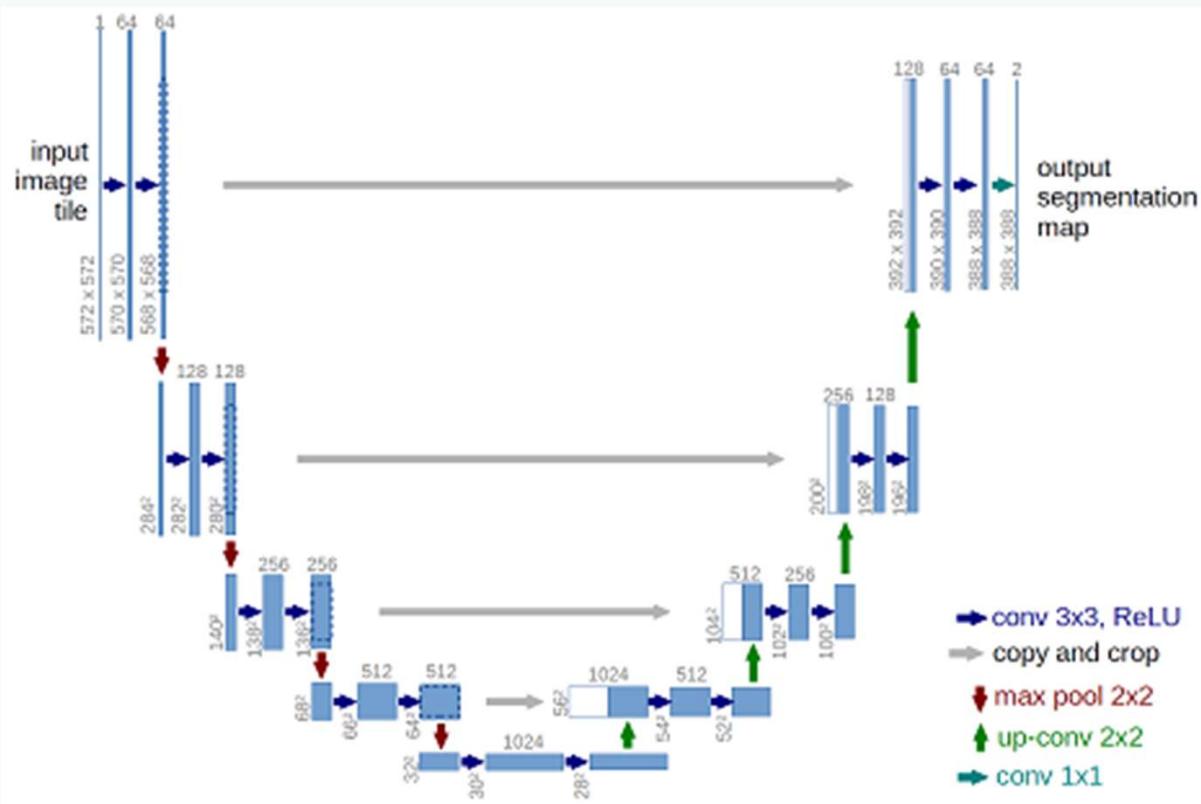
1. Upsampling

Transposed Convolution



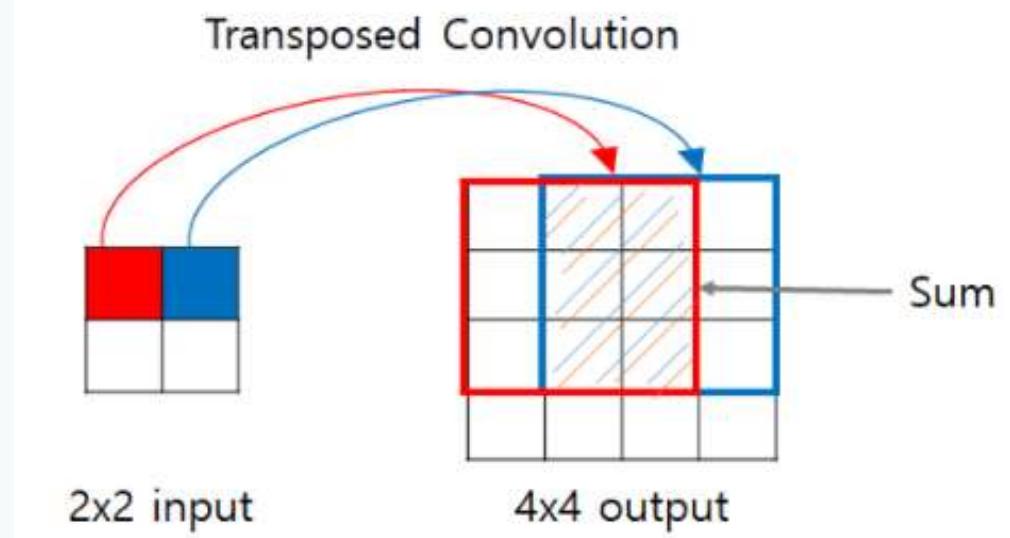
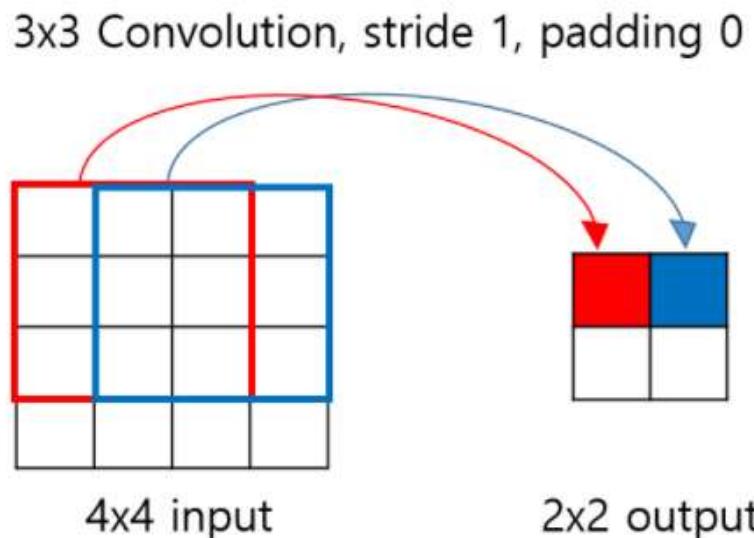
1. Upsampling

unet



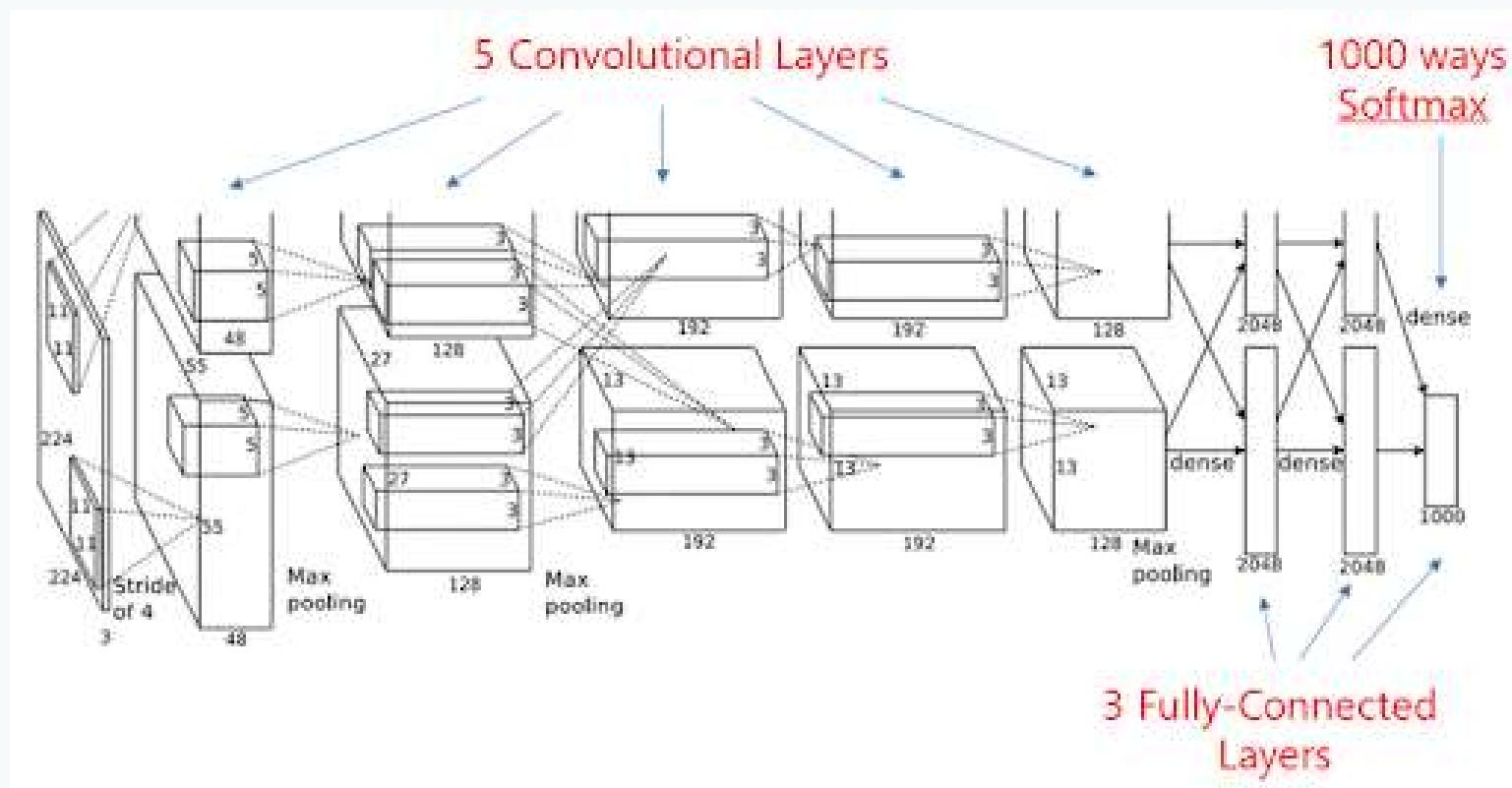
1. Upsampling

Transposed Convolution



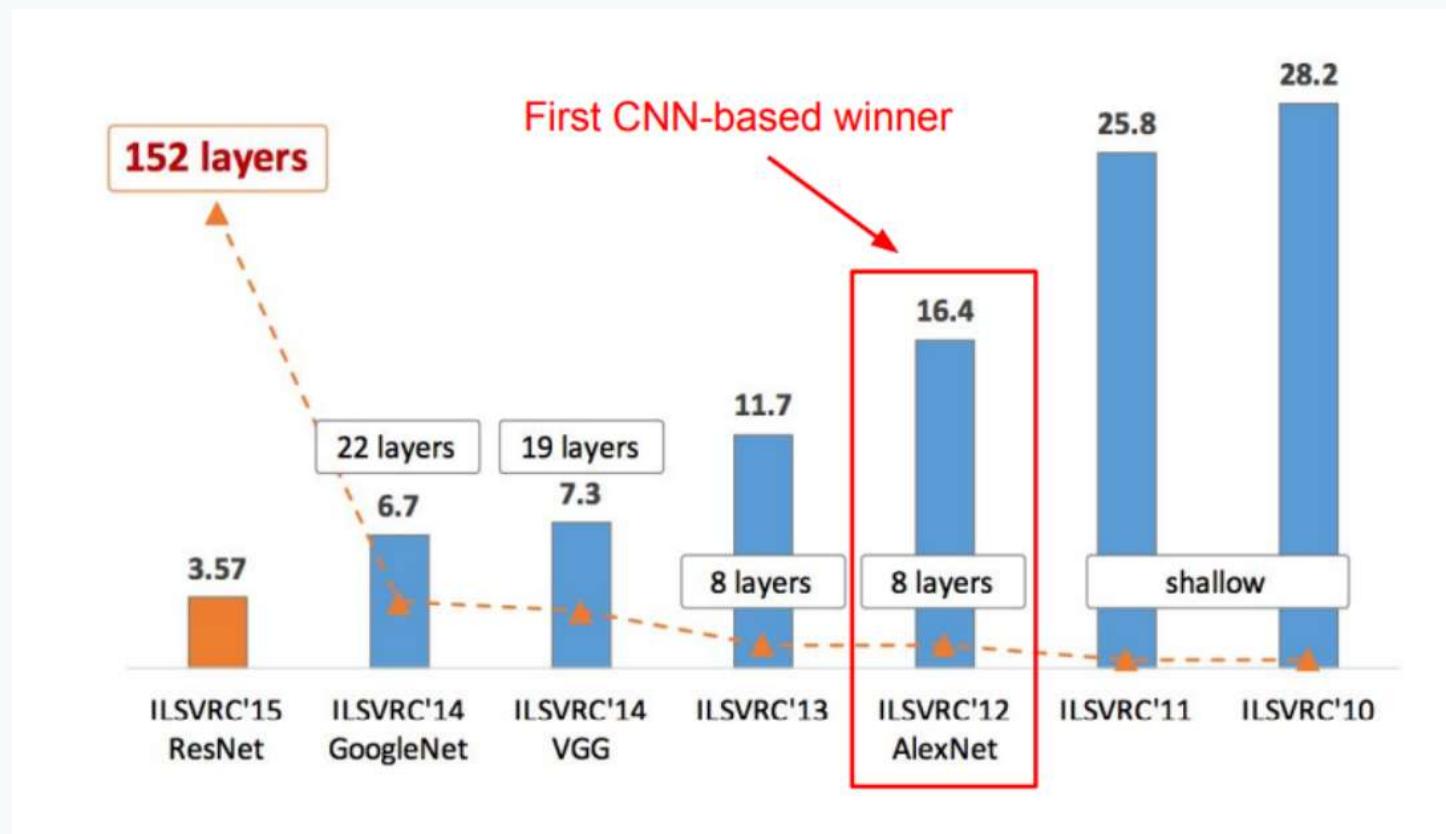
1. CNN 모델 – PAPER REVIEW

Alex NET



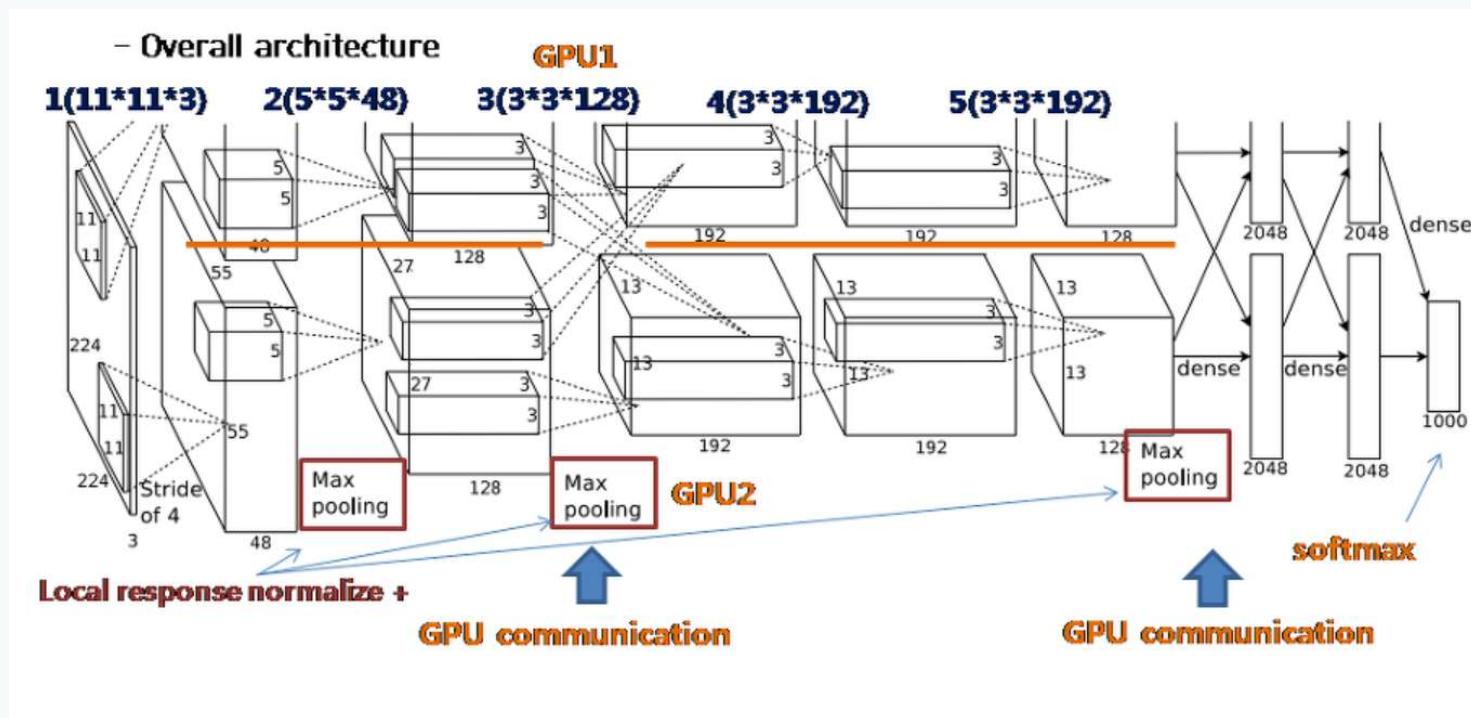
1. CNN 모델 – PAPER REVIEW

Alex NET



1. CNN 모델 – PAPER REVIEW

Alex NET : 구조

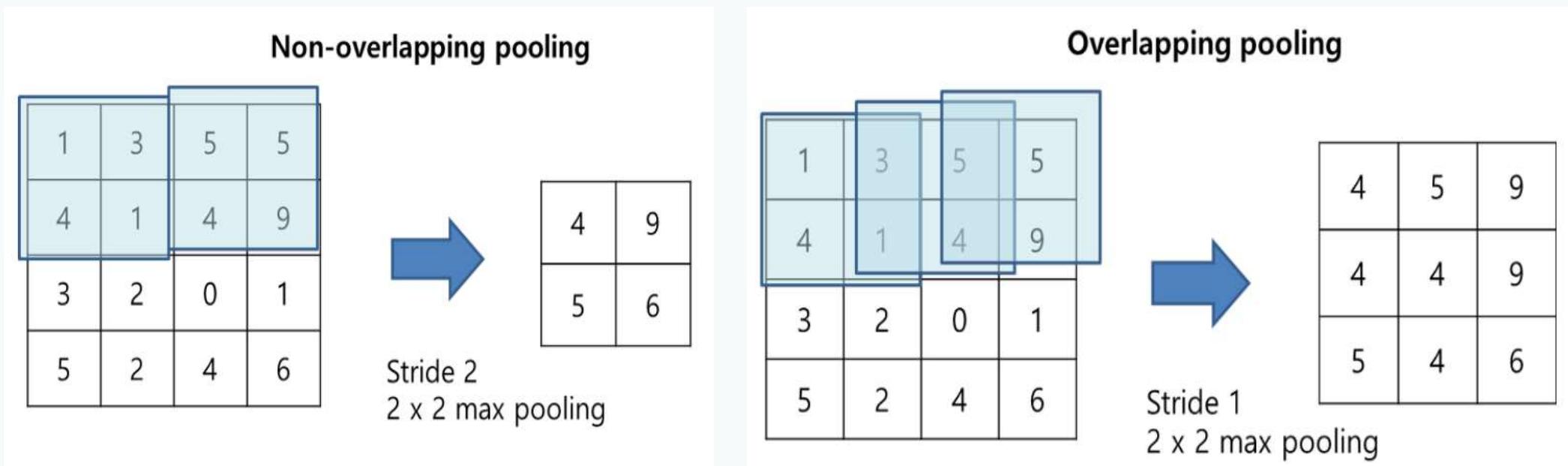


Pooling size =3, stride =2
최초의 max pooling 시도 -> overfitting 막는데 도움을 줌

1. CNN 모델 – PAPER REVIEW

Alex NET : 구조

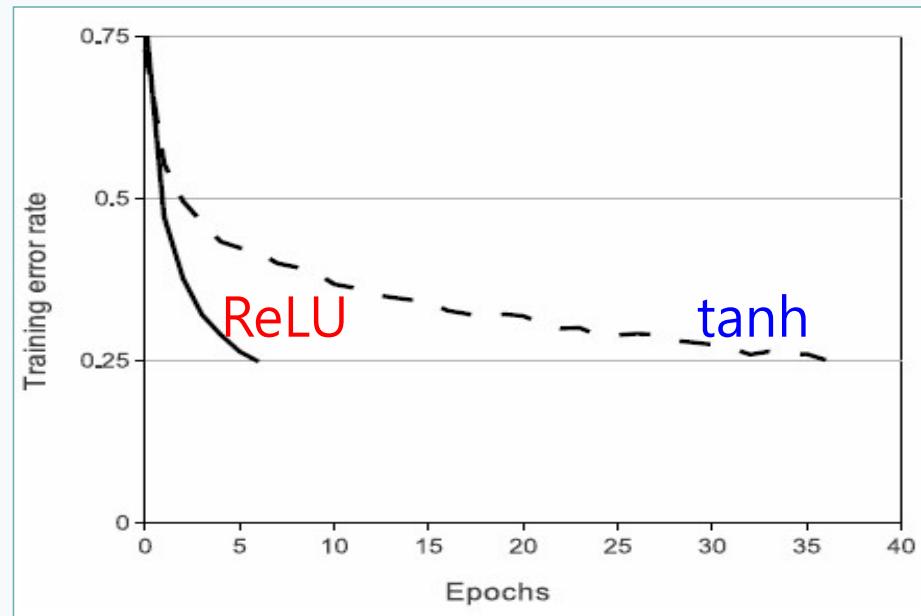
MAX Pooling + Overlapping pooling 사용



1. CNN 모델 – PAPER REVIEW

Alex NET : 구조

Activation Function : Relu



Faster Convergence!

1. CNN 모델 – PAPER REVIEW

Alex NET : 구조

regularization 적용 : Local Response Normalization

The response-normalized activity is given by:

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

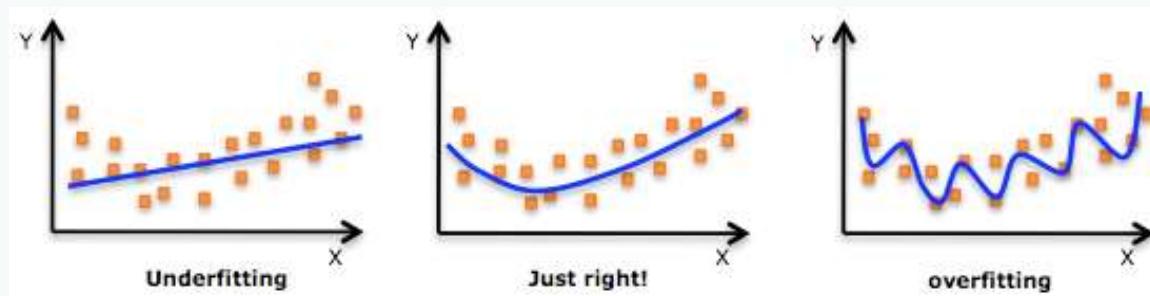
It implements a form of **lateral inhibition** inspired by real neurons.

인접한 kernel map n개를 모아서 정규화. 결과적으로 강한 자극이 약한 자극을 막는 효과(lateral inhibition)을 모방하여 일반화에도 효과적.

1. CNN 모델 – PAPER REVIEW

Alex NET : 구조

오버피팅을 막기 위해 다음의 방법 사용



대표적으로 사용한 방법은 다음 두가지.

- Data augmentation
- Drop out

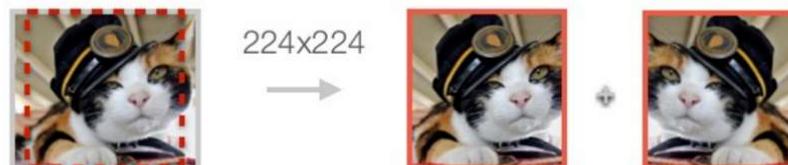
1. CNN 모델 – PAPER REVIEW

Alex NET – Data augmentation

a. No augmentation (= 1 image)



b. Flip augmentation (= 2 images)



c. Crop+Flip augmentation (= 10 images)



총 2048배
학습데이터 증가

1. CNN 모델 – PAPER REVIEW

Alex NET – Data augmentation

RGB pixel에 대한 PCA를 진행하여 augmentation.

color의 변화와 출렁임에도 유지하는 object identiy를 학습하도록 함.

1 단계 : RGB 3채널의 이미지를 R과 G와 B로 분리.

2 단계 : 분리한 각 R, G, B 컬럼에 principal vector, principal eigenvalue,에 약간의 random 요소를 더해 RGB를 미세하게 같은 방향, 크기로 변동

. $\{\mathbf{p}_i\}$: principal eigen vector

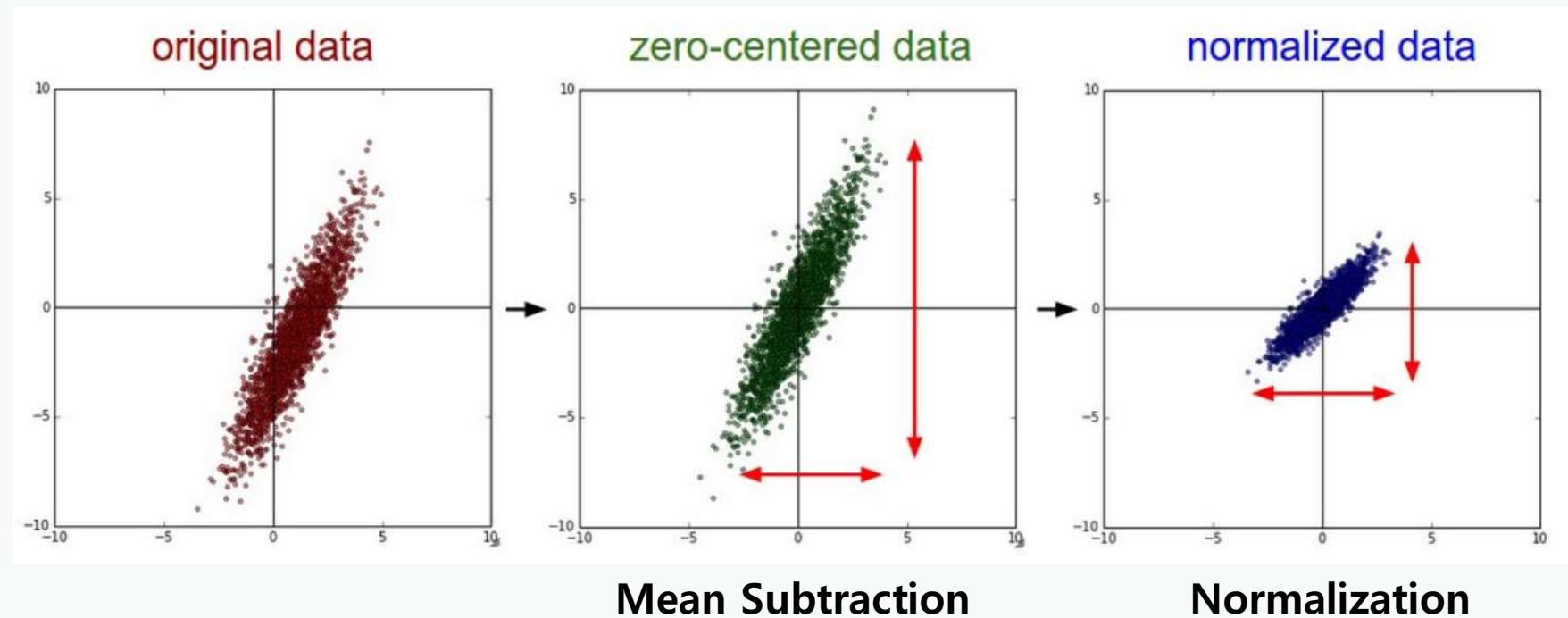
. $\{\alpha_i\} \sim N(0, 0.1)$

. $\{\lambda_i\}$: eigen value

$$[I_{xy}^R, I_{xy}^G, I_{xy}^B]^T = [I_{xy}^R, I_{xy}^G, I_{xy}^B]^T + [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3][\alpha_1 \lambda_1, \alpha_2 \lambda_2, \alpha_3 \lambda_3]^T$$

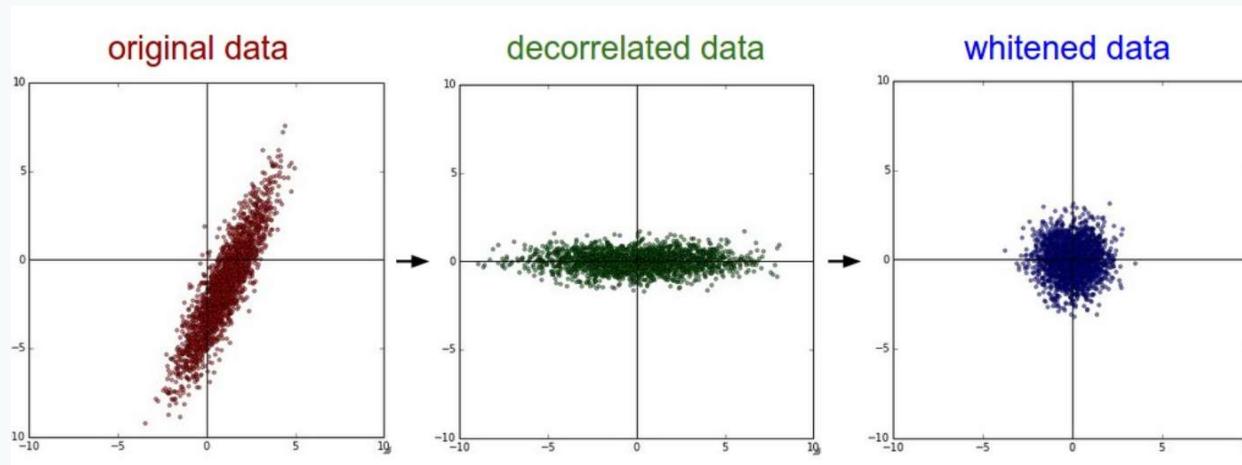
1. CNN 모델 – PAPER REVIEW

Alex NET – Data preprocessing



1. CNN 모델 – PAPER REVIEW

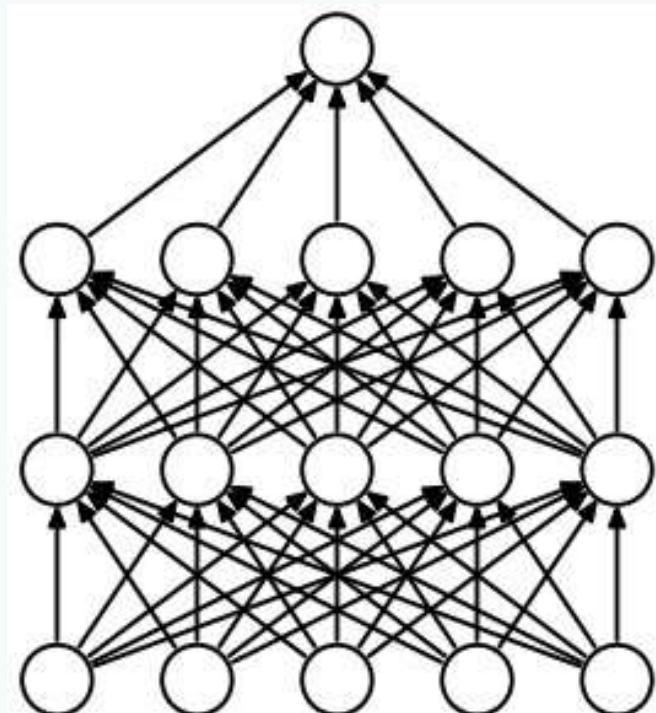
Alex NET – Data preprocessing



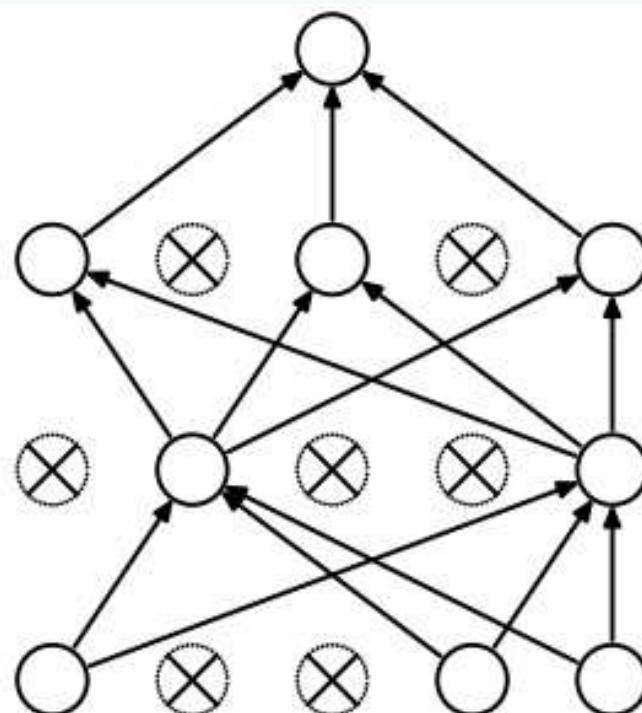
왼쪽은 2차원의 기본 데이터. 중간은 PCA를 적용한 것인데 데이터들이 모두 zero-centered 되어 있고 데이터의 covariance 행렬의 eigen basis로 회전된 것을 볼 수 있음. (covariance 행렬이 diagonal 된 상태이다). 오른쪽은 eigenvalue에 의해 추가적으로 scale 되었으며, 데이터의 covariance 행렬이 identity 행렬로 변환.

1. CNN 모델 – PAPER REVIEW

Alex NET – Dropout



(a) Standard Neural Net

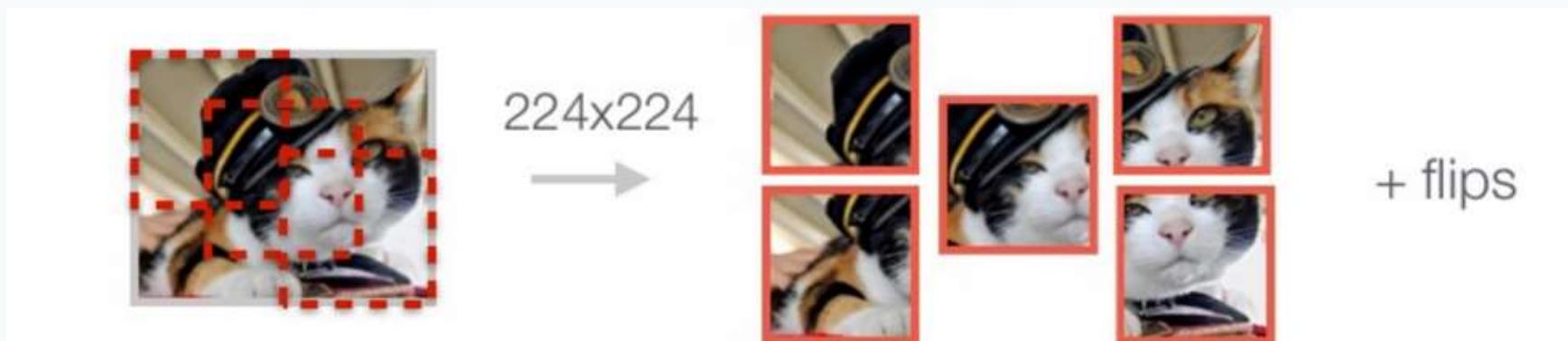


(b) After applying dropout.

1. CNN 모델 – PAPER REVIEW

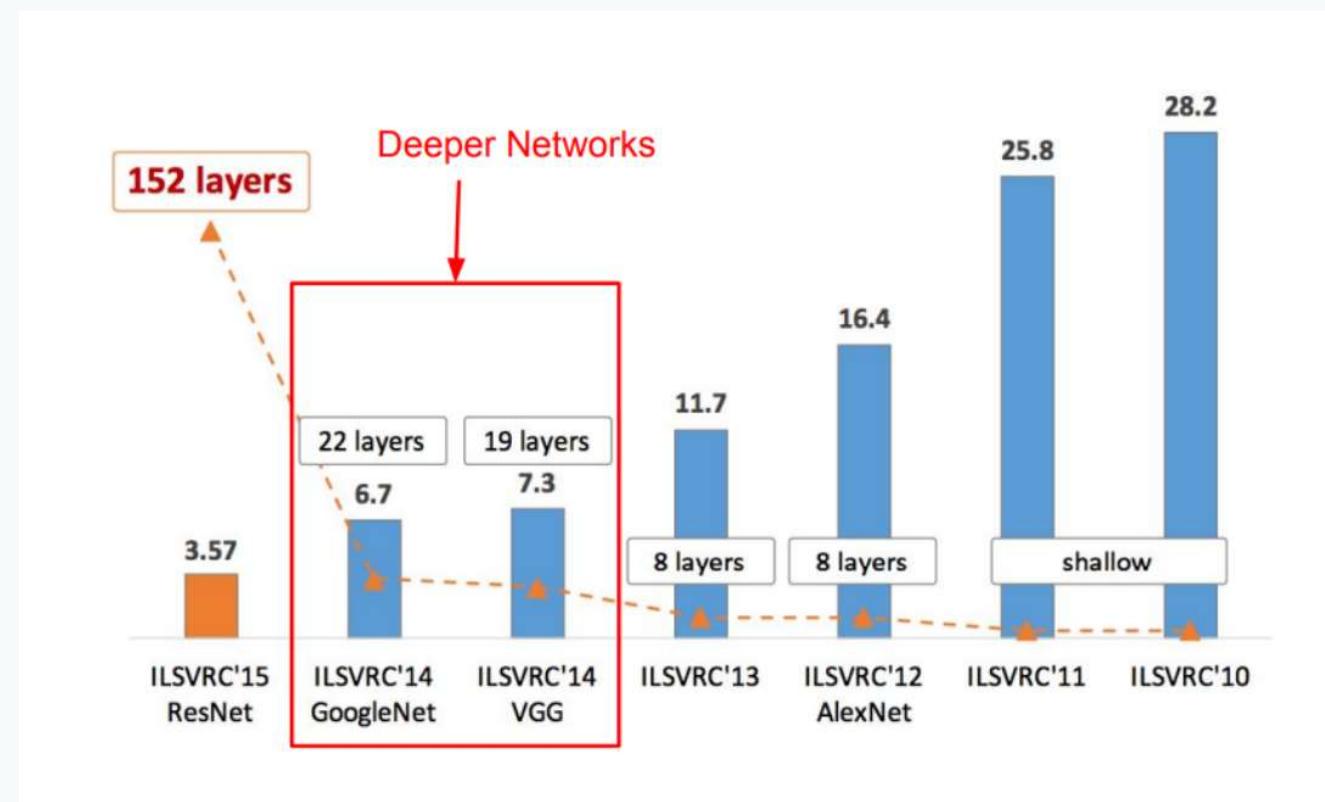
Alex NET – inference (양상을 적용)

테스트할 이미지 한장에서 이미지 4귀퉁이 + 가운데에서 224x224 size로 crop한 5장을 flip하여 10장의 이미지를 만들고 10장을 테스트(inference)하여 결과를 양상을 하여 결과를 나타냄.



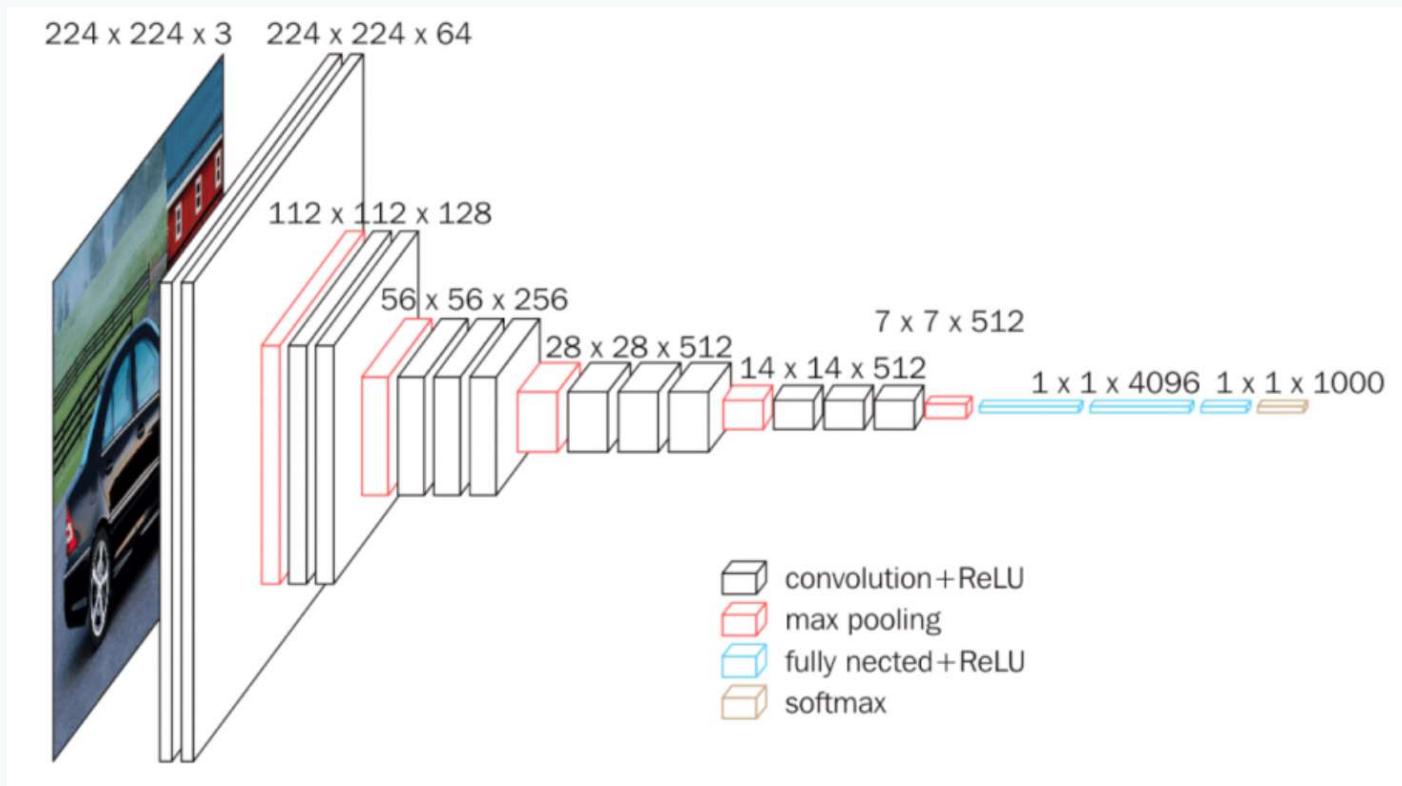
1. CNN 모델 – PAPER REVIEW

VGG NET



1. CNN 모델 – PAPER REVIEW

VGG NET



1. CNN 모델 – PAPER REVIEW

VGG NET

- 목표

네트워크의 깊이를 깊게 만드는 것이 성능에 어떤 영향을 미치는가 확인

- 방법

깊이의 영향만을 확인하기 위해 커널 사이즈는 가장 작은 3x3으로 고정

1. CNN 모델 – PAPER REVIEW

VGG NET - 구조

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

1. CNN 모델 – PAPER REVIEW

VGG NET – 성능 비교

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64	conv3-64	conv3-64	conv3-64	conv3-64
LRN		conv3-64	conv3-64	conv3-64	conv3-64
maxpool					
conv3-128	conv3-128	conv3-128	conv3-128	conv3-128	conv3-128
	conv3-128	conv3-128	conv3-128	conv3-128	conv3-128
maxpool					
conv3-256	conv3-256	conv3-256	conv3-256	conv3-256	conv3-256
conv3-256	conv3-256	conv3-256	conv3-256	conv3-256	conv3-256
maxpool					
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
maxpool					
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512
FC-4096					
FC-4096					
FC-1000					
soft-max					

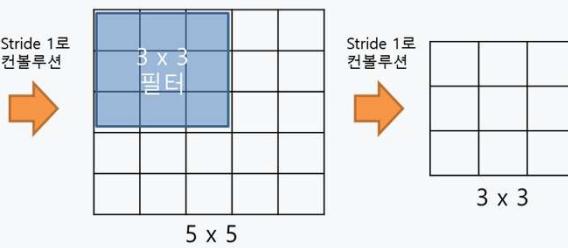
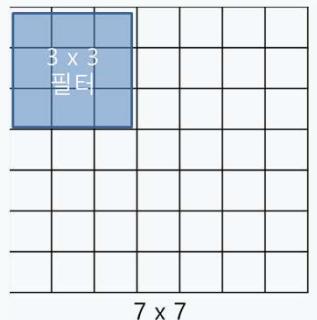
1. A에 LRN을 적용하지 않은 모델
A에 LRN을 적용한 모델 성능을 비교

결과적으로 성능 향상에 효과가 없었음.

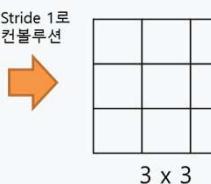
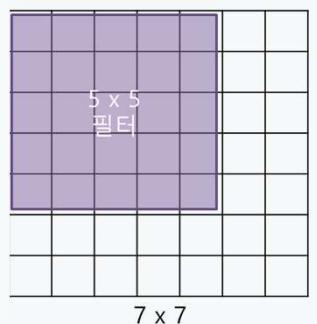
2. 레이어가 깊어질 수록
성능이 좋아지는 것을 확인

1. CNN 모델 – PAPER REVIEW

VGG NET – 3X3 Filter



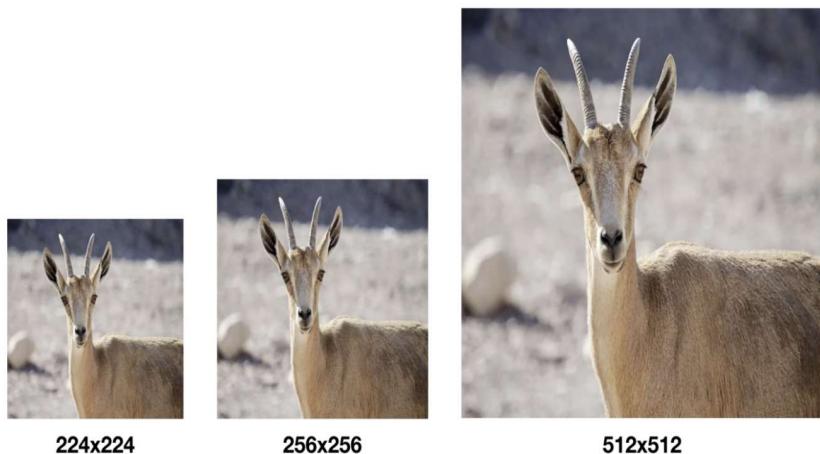
- 여러 개의 ReLU non-linear를 사용할 수 있다.
큰 필터로 구성된 하나의 레이어를 작은 필터의 여러 레이어로 나누었기 때문에 ReLU non-linearity가 들어갈 곳이 더 많아진다.



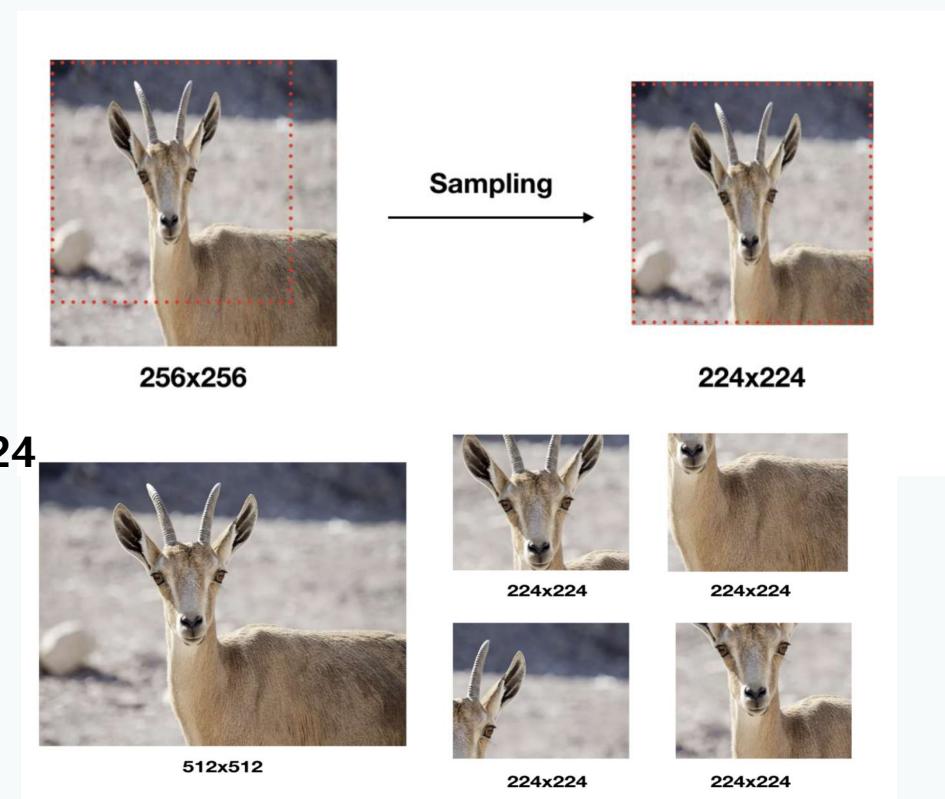
- 학습 해야할 weight의 수가 많이 줄어든다.
regularization(모델의 일반화) 측면에서 큰 이점
학습 속도가 빠름

1. CNN 모델 – PAPER REVIEW

VGG NET – 학습이미지 : 224x224 고정



Crop을 통해 224x224
이미지 구축



다양한 이미지를 위 3개의 사이즈 처럼 구축

1. CNN 모델 – PAPER REVIEW

VGG NET – 학습이미지 : 224x224 고정

Table 3: ConvNet performance at a single test scale.

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
A	256	256	29.6	10.4
A-LRN	256	256	29.7	10.5
B	256	256	28.7	9.9
C	256	256	28.1	9.4
	384	384	28.1	9.3
	[256;512]	384	27.3	8.8
D	256	256	27.0	8.8
	384	384	26.8	8.7
	[256;512]	384	25.6	8.1
E	256	256	27.3	9.0
	384	384	26.9	8.7
	[256;512]	384	25.5	8.0

싱글 스케일 이미지 사이즈만으로 학습

Table 4: ConvNet performance at multiple test scales.

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
B	256	224,256,288	28.2	9.6
	256	224,256,288	27.7	9.2
C	384	352,384,416	27.8	9.2
	[256;512]	256,384,512	26.3	8.2
D	256	224,256,288	26.6	8.6
	384	352,384,416	26.5	8.6
	[256;512]	256,384,512	24.8	7.5
E	256	224,256,288	26.9	8.7
	384	352,384,416	26.7	8.6
	[256;512]	256,384,512	24.8	7.5

멀티 스케일 이미지 사이즈만으로 학습

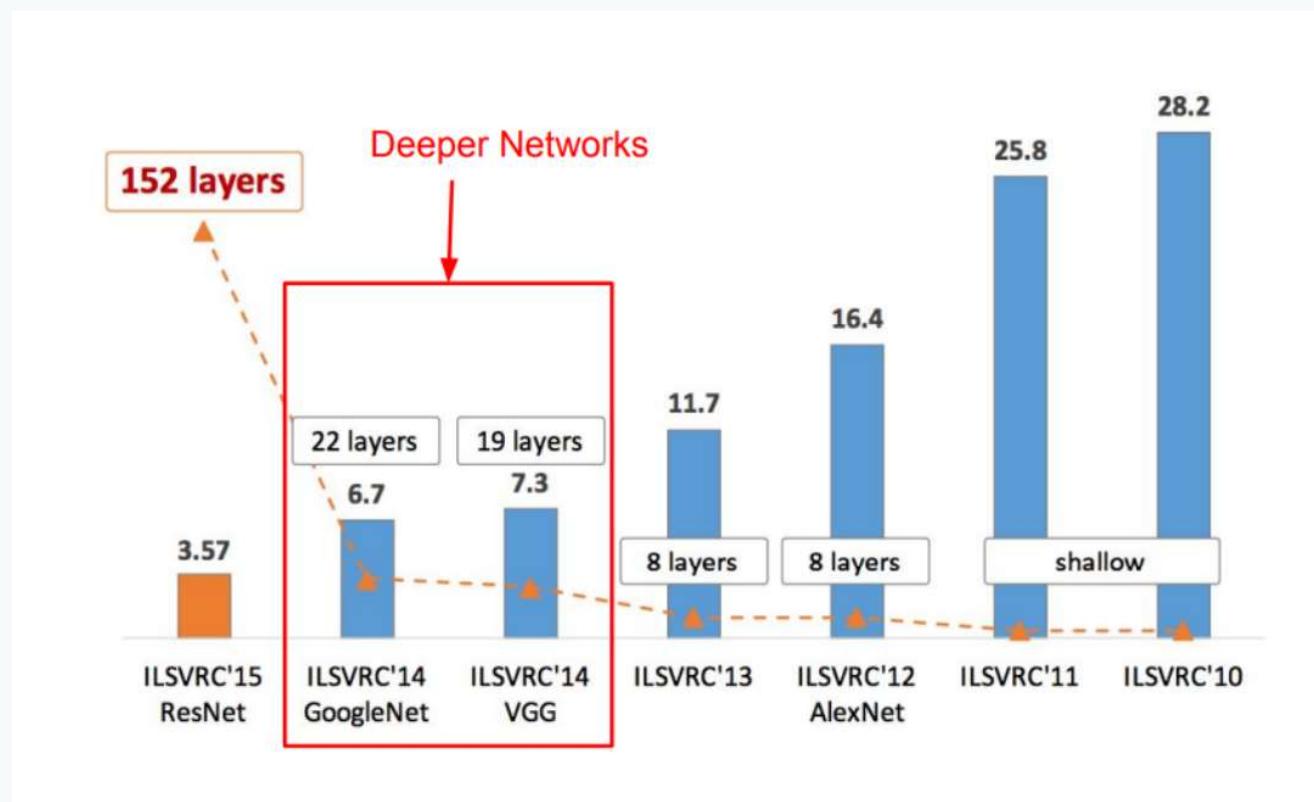
1. CNN 모델 – PAPER REVIEW

VGG NET – 단점

- VGGNet은 간단한 구조를 가졌지만, fully connected layer가 3개가 있고, 풀링(pooling)을 거친 뒤에는 feature map의 개수가 2배로 커지면서 **필요한 파라미터가 과도하게 많음**.
- 파라미터가 많다는 것은 딥러닝의 고질적인 문제인, **gradient vanishing, overfitting 등의 문제가 발생할 가능성이 크다는 의미**.
- 실제로 VGG16과 VGG19는 **학습에 어려움이 있음**.
- 논문 저자들은 이 문제를 해결하기 위해 "A" 모델의 **pretrained model**로 하여 학습.

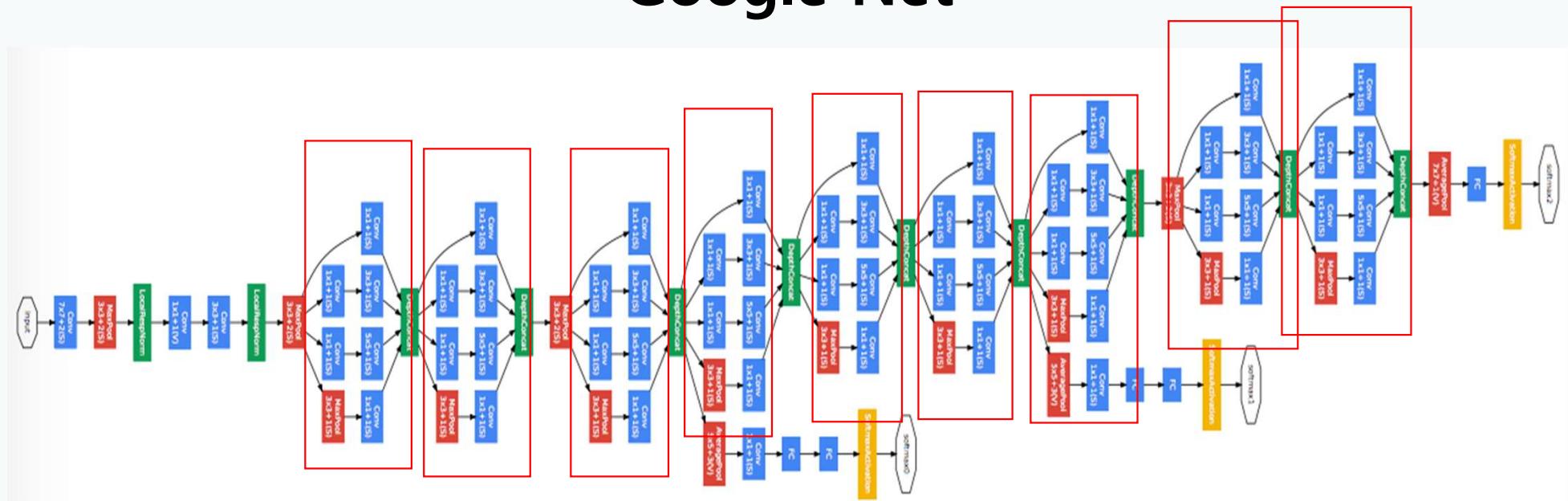
1. CNN 모델 – PAPER REVIEW

Google NET



1. CNN 모델 – PAPER REVIEW

Google Net



- Inception 모듈 사용
- 22개의 layers, 5M parameters(AlexNet의 1/12 수준)

1. CNN 모델 – PAPER REVIEW

Google Net - 컨셉

Layer의 깊이가 깊어질 수록 성능은 높아 질 수 있다. 그래서 Layer를 깊이를 늘려 성능을 높이고자 했다. 그러나 다음과 같은 문제가 발생한다.

1. free parameter 수 증가
2. 데이터 학습양이 작으면 오버 피팅 문제
3. 연산량이 늘어나게 된다. (필터의 갯수가 증가하면 연산량은 제곱 증가)
4. Vanishing Gradient 문제

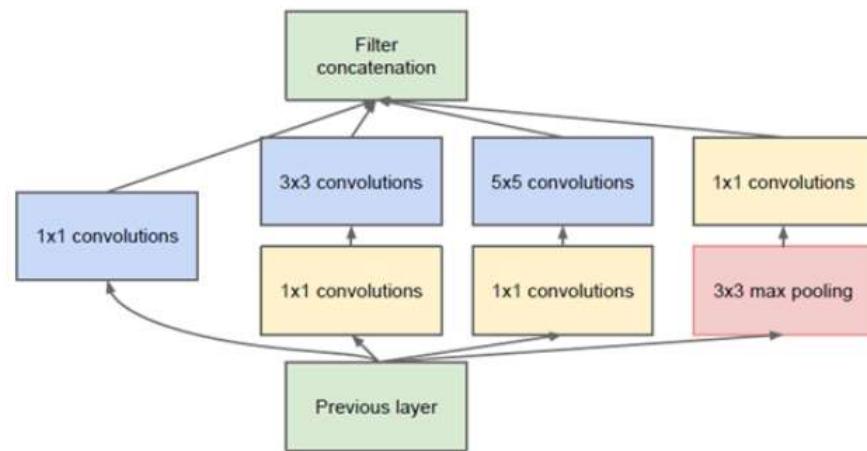
즉, 학습은 잘되었으나 테스트할때 성능이 나오지 않을 수 있다.

또한, 모델이 너무 커져 학습과 테스트에 많은 시간을 소비하게 된다.

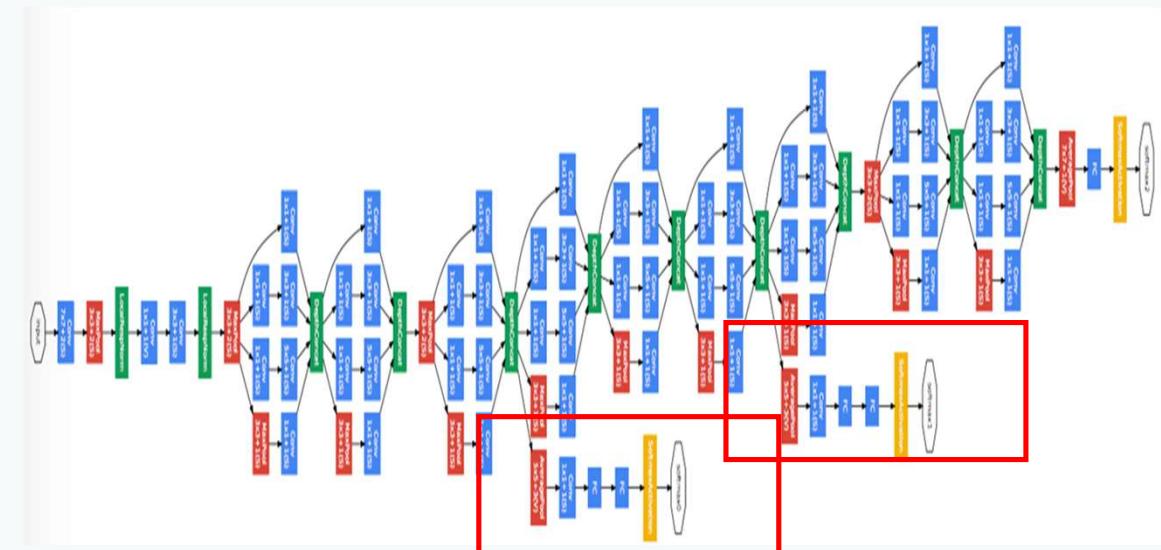
1. CNN 모델 – PAPER REVIEW

Google Net – 모델 솔루션

Inception module



auxiliary classifier



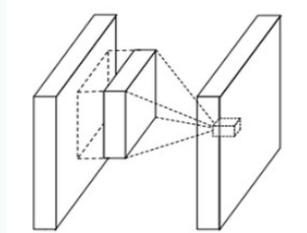
1. CNN 모델 – PAPER REVIEW

Google Net – motivation

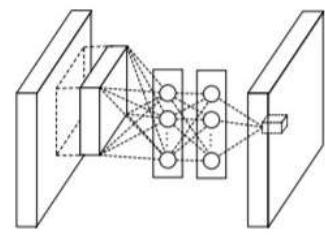
NIN(Network In Network)

일반적인 CNN 은 Feature Extraction(conv + pooling) + classifier(fully connected NN)로 구성.

이러한 구성은 Local receptive field에서 feature를 추출 능력은 우수하지만 비선형 성질의 feature 추출은 한계.



CNN(Linear conv)



Mlp conv

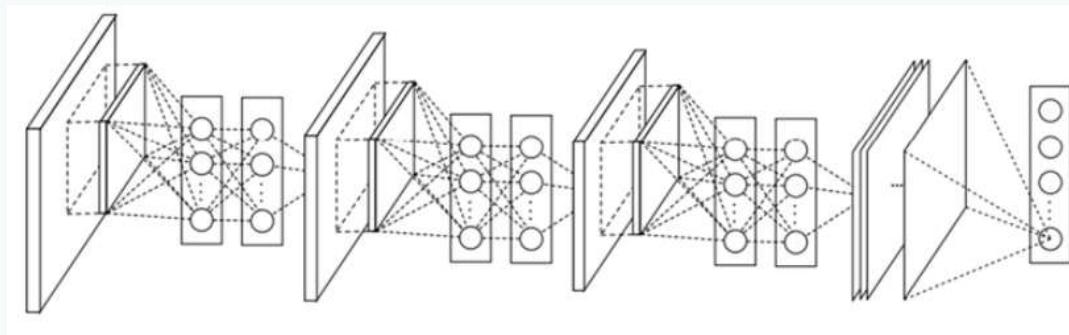
NN의 경우 비선형적인 특징을 잘 추출하는 특징이 있음.

이것을 활용하여 비선형 특징을 잘 추출하도록 구조화 함.

1. CNN 모델 – PAPER REVIEW

Google Net – motivation

MLP Conv의 특징

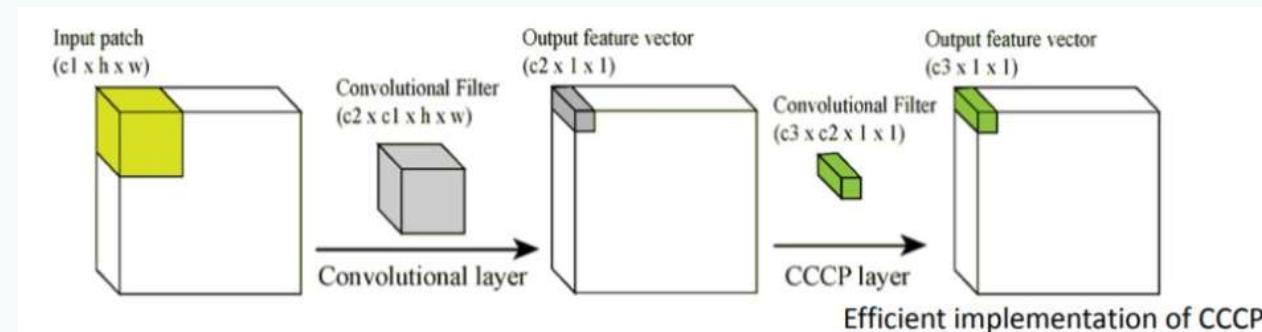


- MLP를 여러 개로 쌓아서 구조.
- classifier에 fully connected NN가 없음.
- fully connected NN 대신에 Global average pooling을 사용.
- 효과적인 feature-vector를 추출하였기 때문에 pooling만으로도 충분하다고 주장.
- 때문에, classifier에서 average pooling만 사용하였기 때문에 overfitting의 문제를 해결.
- 연산량도 대폭 줄어듦.

1. CNN 모델 – PAPER REVIEW

Google Net – motivation

NIN 구조화를 위한 1×1 conv 활용



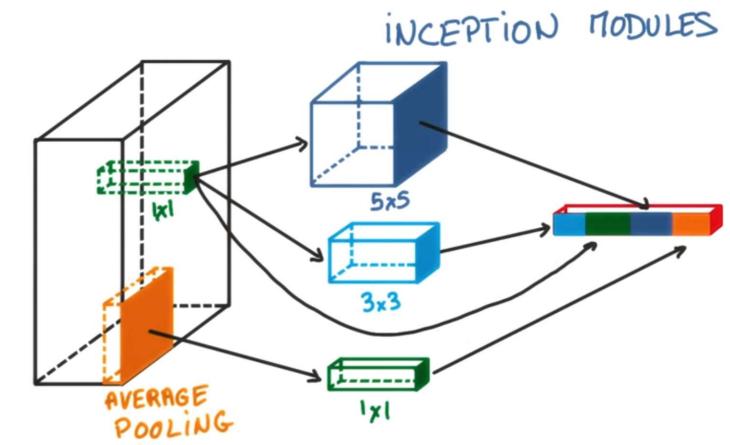
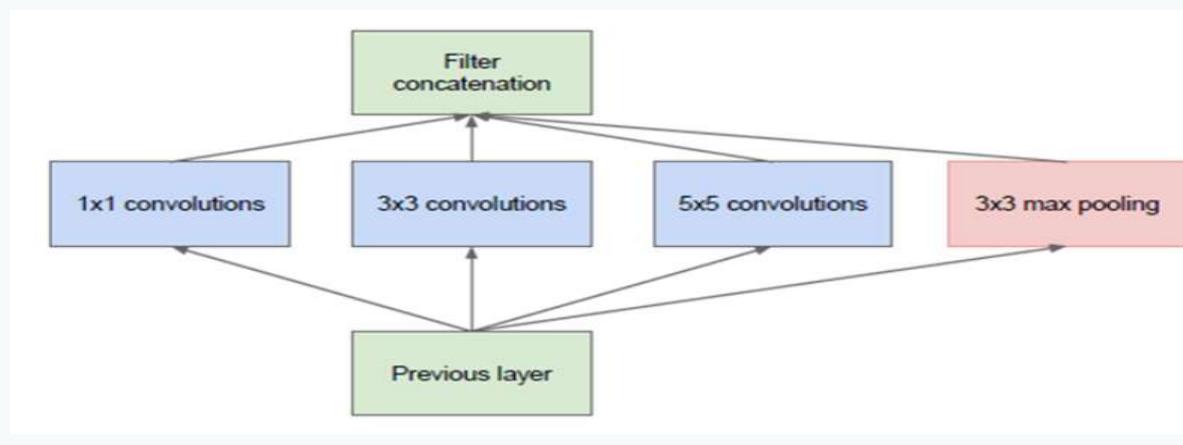
NIN은 일반적인 convolution 연산 후 MLP를 대신하여 1×1 convolution 연산.

다음과 같은 이점을 얻을 수 있다.

- MLP와 같은 효과를 얻을 수 있다 (비선형 특징점 추출)
- 차원 축소.(30x30의 피쳐맵 10개를 30x30 피쳐맵 5개로 줄일 수 있음)

1. CNN 모델 – PAPER REVIEW

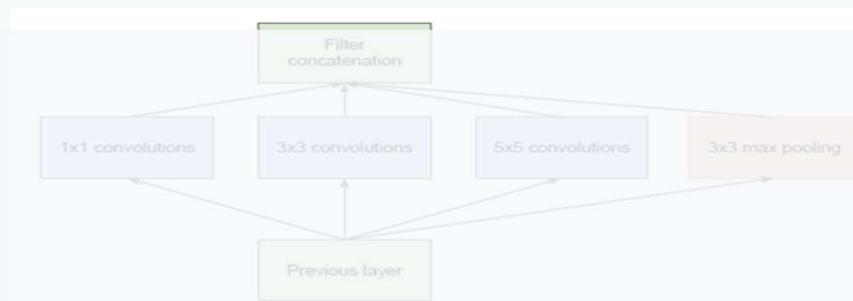
Google Net – Inception module



- 1x1 conv, 3x3 conv , 5x5 conv , 3x3 max pooling을 나란히 놓아 다양한 scale feature를 추출하도록 고안.
(Local receptive field에서 더 많은 featur을 추출하기 위해 여러 개의 convolution을 적용)
- feature 추출 등의 과정은 최대한 sparse함 을 유지.
- 행렬 연산 자체는 이들을 합쳐 최대한 dense하게 만듦.

1. CNN 모델 – PAPER REVIEW

Google Net – Inception module

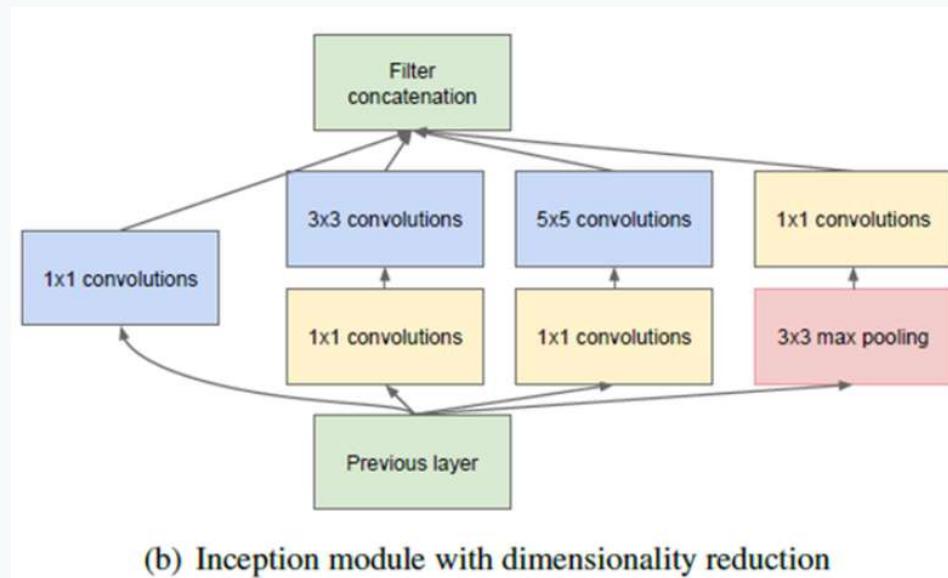


엄청난 컴퓨터 파워를 요구하게 되는 구조를 가짐

- 1x1 conv, 3x3 conv , 5x5 conv , 3x3 max pooling을 나란히 놓아 다양한 scale feature를 추출하도록 고안.
(Local receptive field에서 더 많은 featur을 추출하기 위해 여러 개의 convolution을 적용)
- feature 추출 등의 과정은 최대한 sparse함 을 유지.
- 행렬 연산 자체는 이들을 합쳐 최대한 dense하게 만듦.

1. CNN 모델 – PAPER REVIEW

Google Net – Inception module



- 1x1 conv를 사용하여 차원을 축소한다.
(결국 NIN의 형태와 비슷 – 비선형 feature와 차원축소에 대한 장점을 모두 활용하게 됨)

1. CNN 모델 – PAPER REVIEW

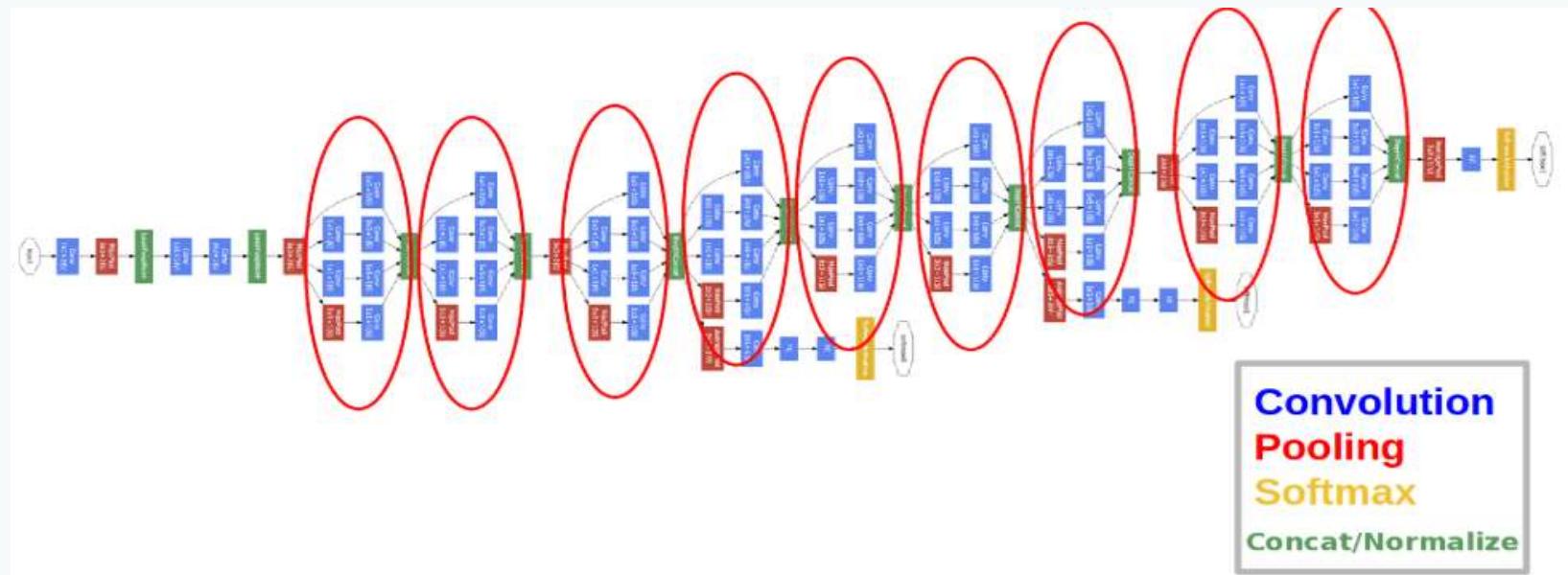
Google Net – Inception module

inception module 요약

- Googlenet의 핵심 설계 목표는 주어진 하드웨어 자원을 최대한 효율적으로 이용하면서도 학습능력은 극대화 할 수 있는 깊고 넓은 망을 갖는 구조를 설계하는 것.
- 인셉션 모듈에 있는 다양한 크기의 conv kernel(파란 그림) 을 통해 다양한 scale의 feature을 효과적 추출.
- 1x1 conv (노란 그림)을 통해 연산량을 크게 경감시켰음.
- 결과적으로 망의 넓이와 깊이를 증가시키게 되었음.
- 인셉션 모듈을 통해 NIN 구조를 갖는 DEEP한 CNN을 설계할 수 있게 되었음

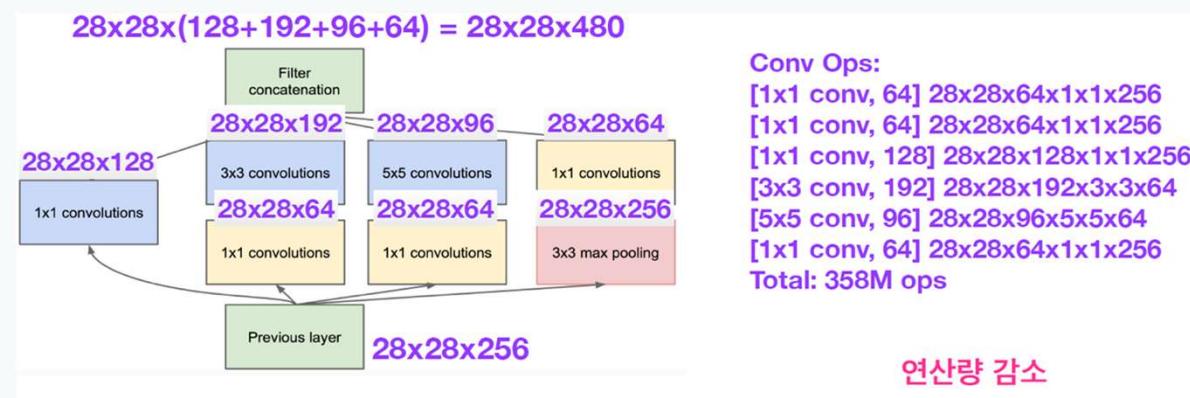
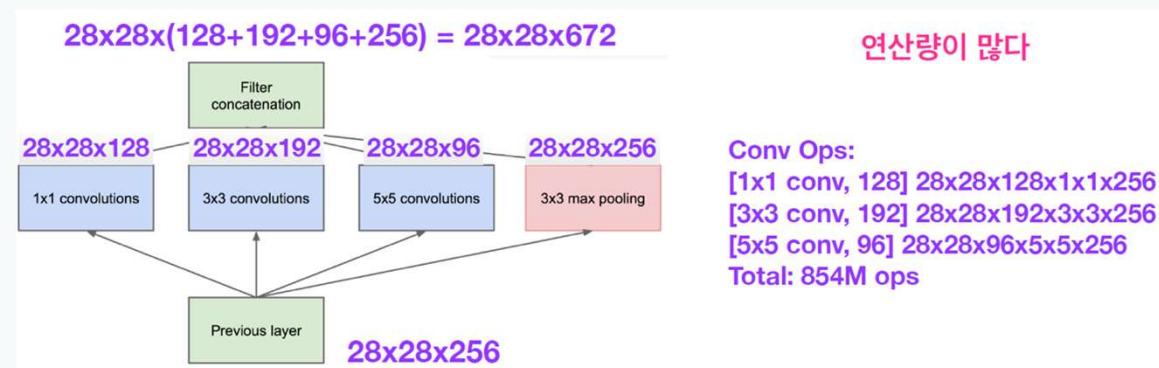
1. CNN 모델 – PAPER REVIEW

Google Net – Inception module



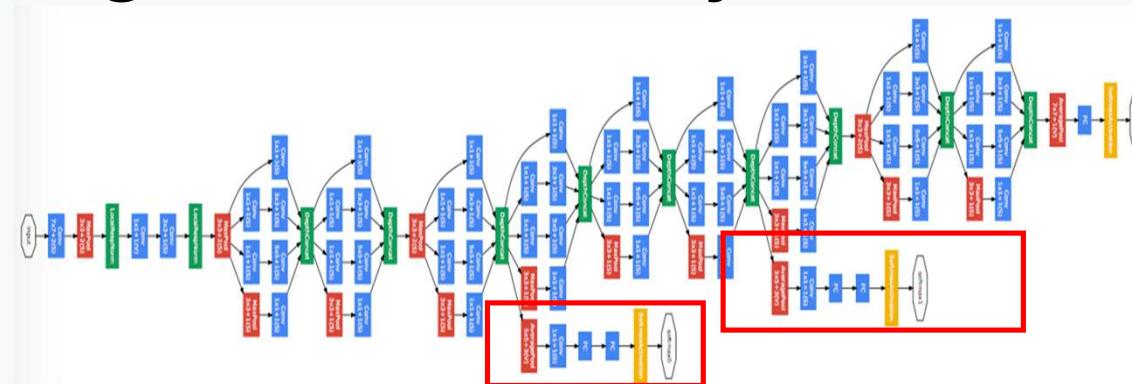
1. CNN 모델 – PAPER REVIEW

Google Net – Inception module



1. CNN 모델 – PAPER REVIEW

Google Net – auxiliary classifier



- 엄청나게 깊은 네트워크에서 Vanishing Gradient 문제가 발생
- auxiliary classifier를 덧붙임. 결과적으로 Loss를 맨 끝 뿐만 아니라 중간 중간에서 구하기 때문에 기울기가 적절하게 역전파 될 수 있음.
- 대신 지나치게 영향을 주는 것을 막기 위해 auxiliary classifier의 loss는 0.3을 곱함.
- 물론 실제로 테스트하는 과정에서는 auxiliary classifier를 제거하고 맨 끝, 제일 마지막의 softmax만을 사용.

1. CNN 모델 – PAPER REVIEW

Google Net – 구조

type	patch size/ stride	output size	depth	#1x1	#3x3 reduce	#3x3	#5x5 reduce	#5x5	pool proj	params	ops
convolution	7x7/2	112x112x64	1							27K	34M
max pool	3x3/2	56x56x64	0								
convolution	3x3/1	56x56x192	2		64	192				112K	360M
max pool	3x3/2	28x28x192	0								
inception (3a)		28x28x256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28x28x480	2	128	128	192	32	96	64	380K	304M
max pool	3x3/2	14x14x480	0								
inception (4a)		14x14x512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14x14x512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14x14x512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14x14x528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14x14x832	2	256	160	320	32	128	128	840K	170M
max pool	3x3/2	7x7x832	0								
inception (5a)		7x7x832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7x7x1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7x7/1	1x1x1024	0								
dropout (40%)		1x1x1024	0								
linear		1x1x1000	1							1000K	1M
softmax		1x1x1000	0								

path size / stride : 커널의 크기와 stride 간격을 말함. Conv에서 7x7/2의 이미는 7x7 size filter를 2픽셀 간격으로 적용

Output size : 얻어지는 feature map의 크기 및 개수를 의미. 112x112x64의 의미는 224x224 크기의 이미지에 2픽셀 간격으로 7x7 filter를 적용하여 총 64개의 featuremap을 추출

Depth : 연속적인 conv layer의 개수를 의미한다. 첫번째 conv layer는 depth가 1이고, 두번째와 인셉션이 적용되어 있는 부분은 모두 2로 되어 있는 이유는 2개의 conv를 연속적으로 적용하기 때문

1x1 : 1x1 conv를 의미. 뒤의 숫자는 feature map의 개수를 의미. 첫번째, 인셉션 3(a)의 1x1위치에 있는 숫자가 64인데 이것은 이전 layer의 192개의 feature map을 받아 64개의 featuremap으로 차원을 축소 시킨 것을 의미.

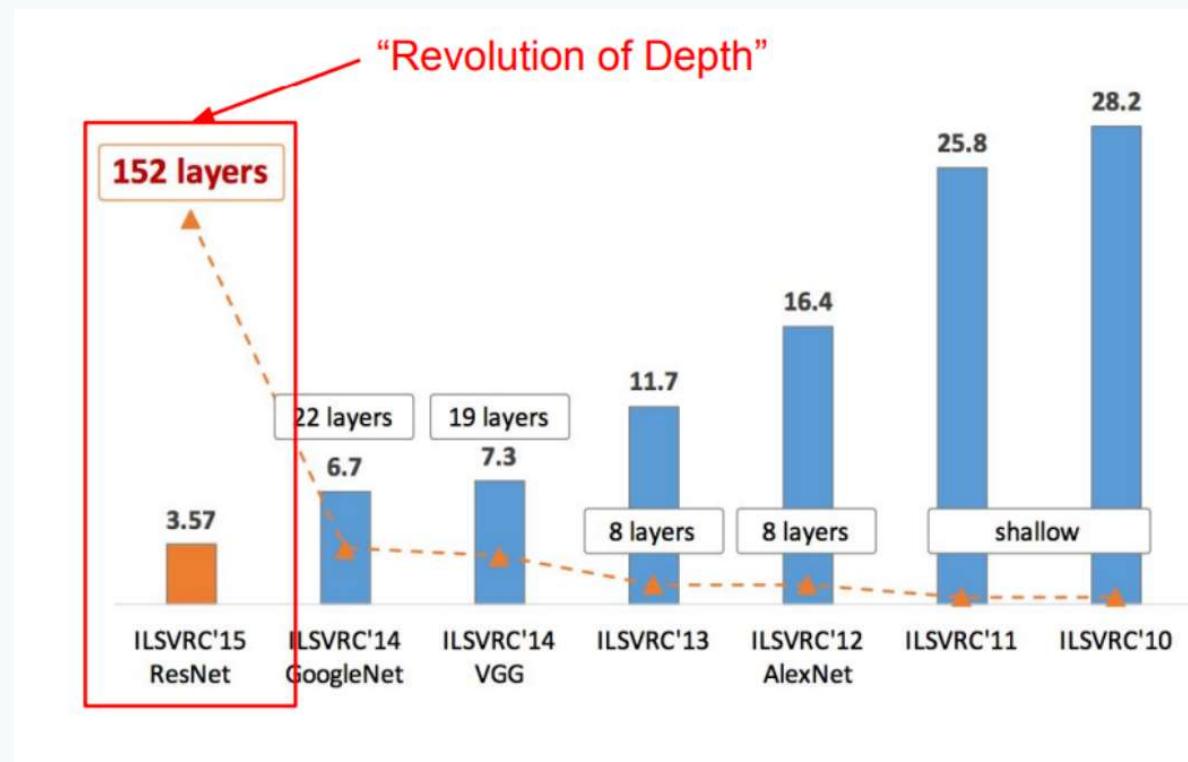
Pool/proj ; max pooling과 max pooling 뒤에 오는 1x1 conv를 적용한것을 의미. 인셉션 3(a) 열의 숫자 32는 max pooling과 1x1 conv를 거쳐 총 32개의 feature map을 추출

인셉션 3(a)에는 256이라는 숫자가 적혀 있는데, 이것은 인셉션 3(a)를 통해 총 256개의 feature map이 만들어 졌다는 뜻.

1x1 conv를 통해 64개 1x1과 3x3연속 conv를 통해 128개 1x1과 5x5 연속 conv를 통해 32개 max pooling과 1x1 conv를 통해 32개 -> 총 256개의 feature map 추출

1. CNN 모델 – PAPER REVIEW

ResNET



1. CNN 모델 – PAPER REVIEW

Res Net

Alex net , VGG Net , Google NET 등

모델을 깊게 쌓아서 성능을 높이고자 하였음.

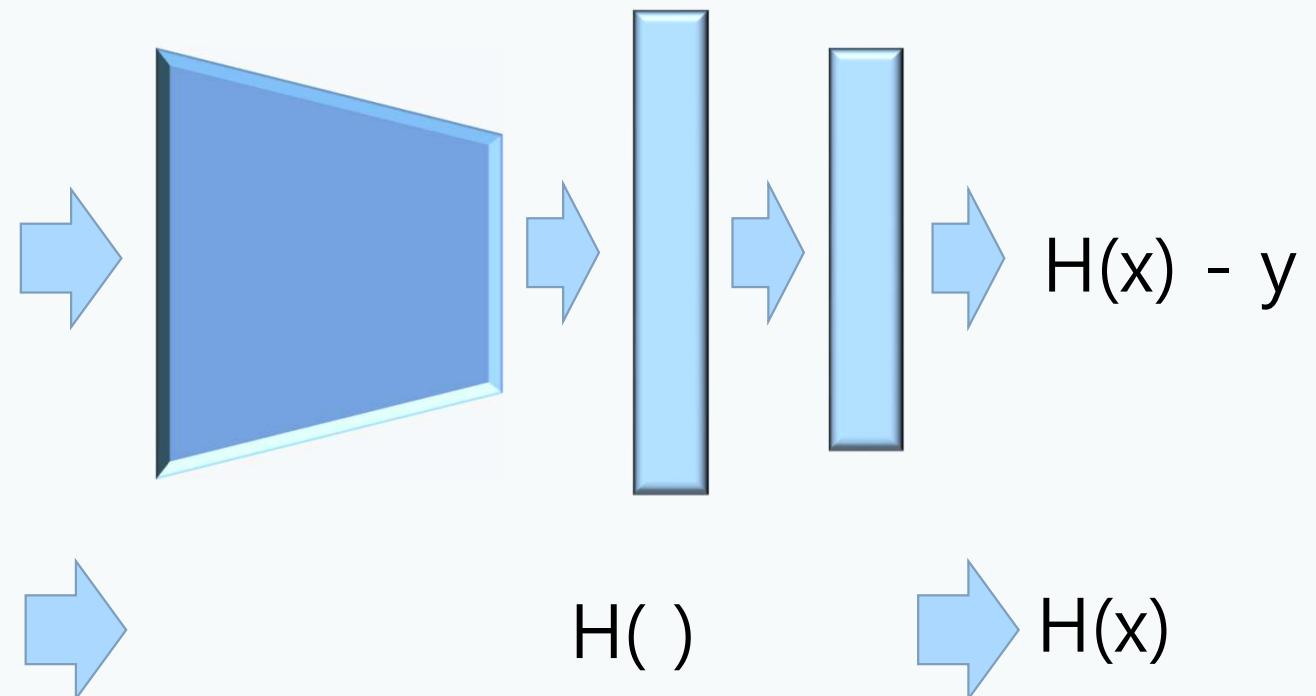
하지만, 깊게 쌓았을때 학습이 어려운 것이 있음.

NN과 같이 여전히 오버피팅, 기울기 소실 등의 문제가 발생

Resnet은 이것을 해결하는 구조 만듦.

1. CNN 모델 – PAPER REVIEW

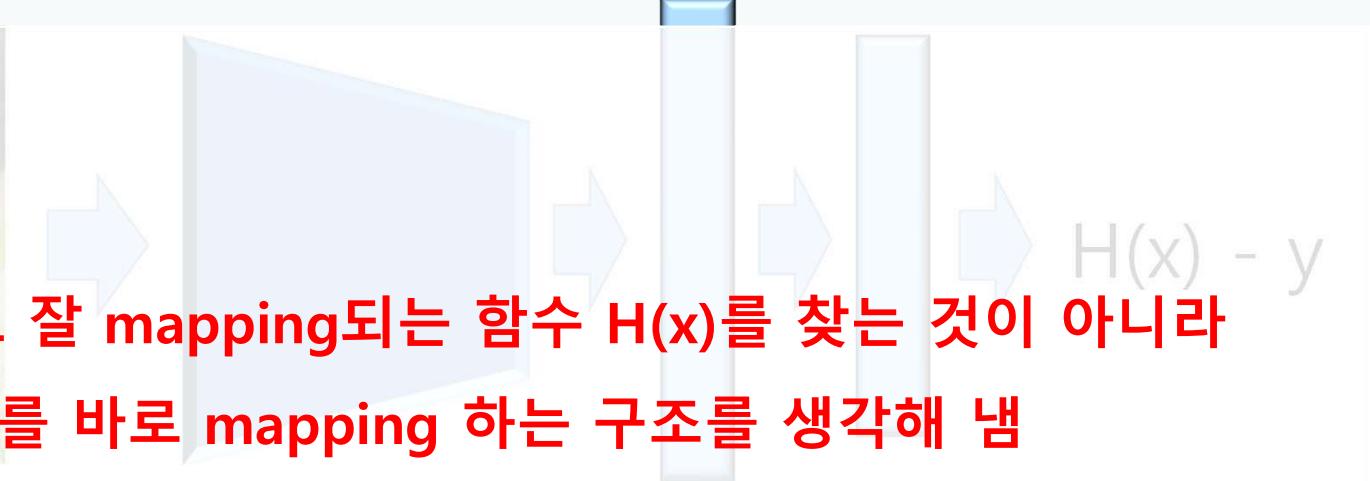
Res Net - Idea



입력 x 가 딥러닝 모델 $H(\cdot)$ 를 거쳐 나온 $H(x)$ 가 정답 y (1 or 0) 을
나타낼 수 있도록 mapping 되도록 $H(\cdot)$ 모델을 학습

1. CNN 모델 – PAPER REVIEW

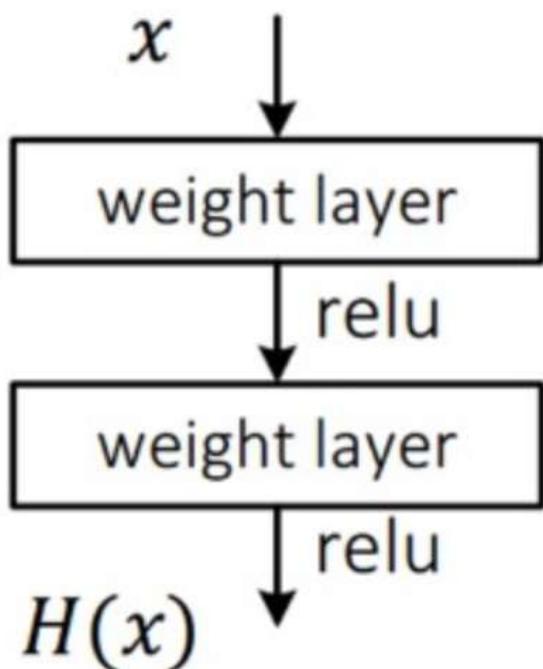
Res Net - Idea



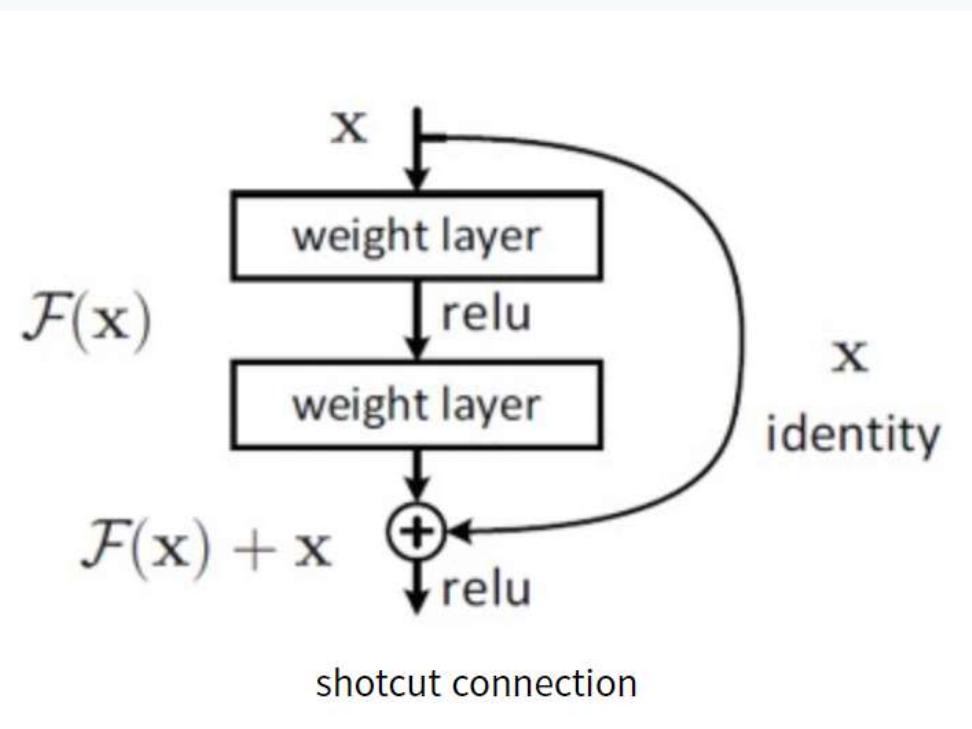
입력 x 가 딥러닝 모델 $H()$ 를 거쳐 나온 $H(x)$ 가 정답 y (1 or 0) 을 나타낼 수 있도록 mapping 되도록 $H()$ 모델을 학습

1. CNN 모델 – PAPER REVIEW

Res Net - Idea



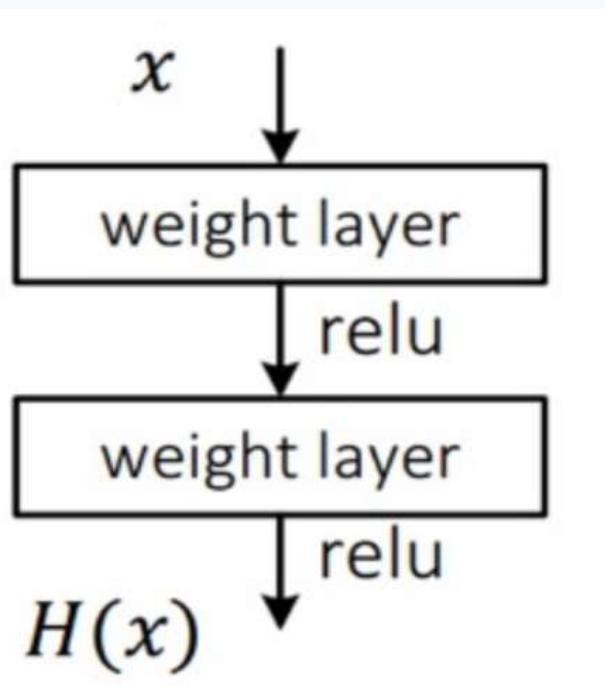
기존 CNN 구조



Resnet 구조

1. CNN 모델 – PAPER REVIEW

Res Net - Idea



기존 CNN 구조

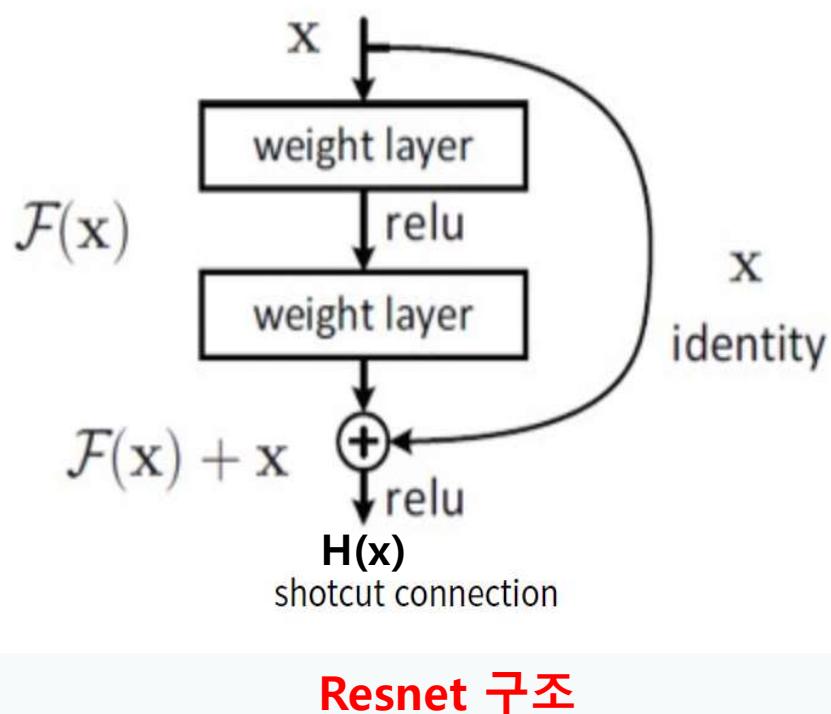
$x \rightarrow y$ 로 mapping을 해야 한다는 idea.

결국 최종 출력이 $H(x) = x$ 가 되어야 함.

그러므로 $H(x)-x= 0$ 되도록 웨이트 학습.

1. CNN 모델 – PAPER REVIEW

Res Net - 구조



$X \rightarrow y$ 로 mapping을 해야 한다는 idea.

$$F(x) = H(x) - x$$

를 미분하면 $H(x)$ 는 전달 전달 되기 때문에

미분하면 기울기 소실이 될 수 있으나

x 팀은 항상 미분 1의 값을 가짐.

저러한 수식이 되려면 왼쪽처럼
때문에 기울기 소실 문제가 해결 됨.
shortcut connection(skip layer)를 해주면 팀.

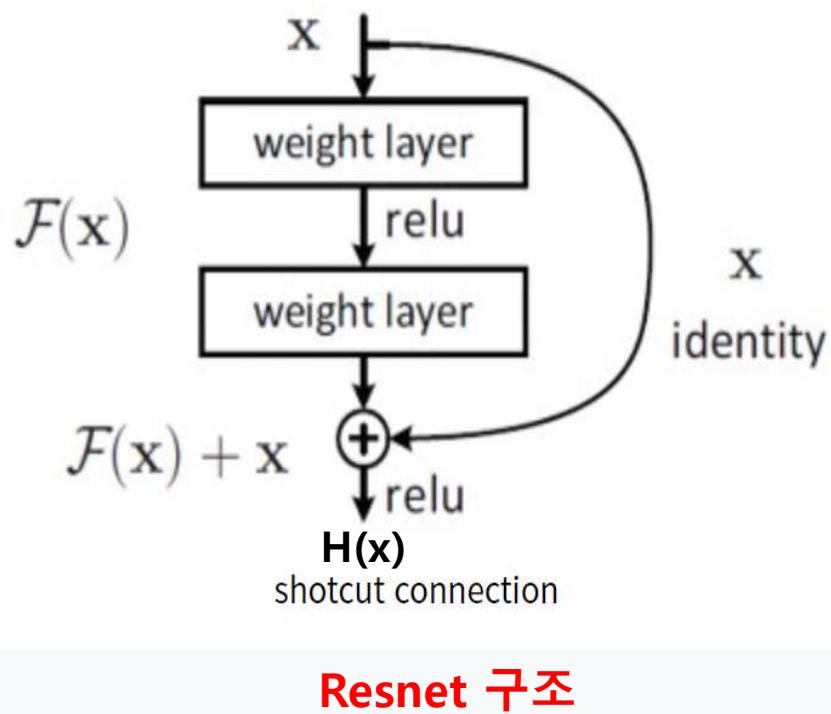
구조적으로 보면 역전파할때 스kip 레이어쪽으로

오차는 그대로 전파되짐 > 잔차라고 함.

이 잔차를 학습하는 것을 Residual Learning

1. CNN 모델 – PAPER REVIEW

Res Net - Idea



$X \rightarrow y$ 로 mapping을 해야 한다는 idea.

결국 최종 출력이 $H(x) = x$ 가 되어야 함.

1. 이미지에서는 $H(x) = x$ 가 되도록 학습
2. 웨이트 레이어 출력 $F(x)$ 는 0이 되도록 학습.
3. $F(x) + x = H(x) = x$ 가 되도록 학습
4. 모든 layer에서 기울기가 $1+F(x)'$ 이므로 기울기 소실을 막아줌

여기서, 미분해도 $F(x)' + 1$ 이므로 기울기 1은 항상 존재

Shotcut connection(=skip layer)를 해주면 됨

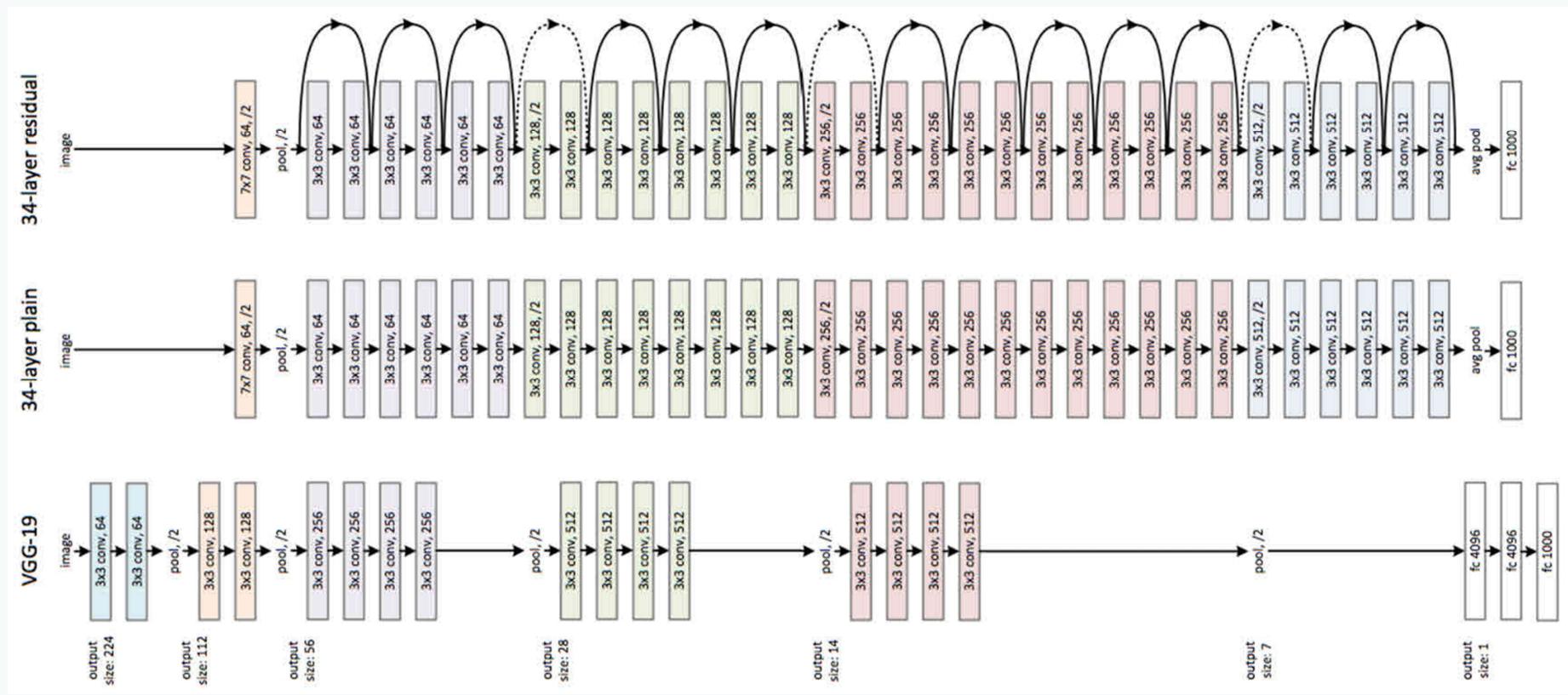
$F(x) + x = H(x)$

$F(x) = H(x) - x \rightarrow$ 잔차라고 함.

이 잔차를 학습하는 것을 Residual Learning

1. CNN 모델 – PAPER REVIEW

Res Net - 구조



1. CNN 모델 – PAPER REVIEW

DenseNet 구조

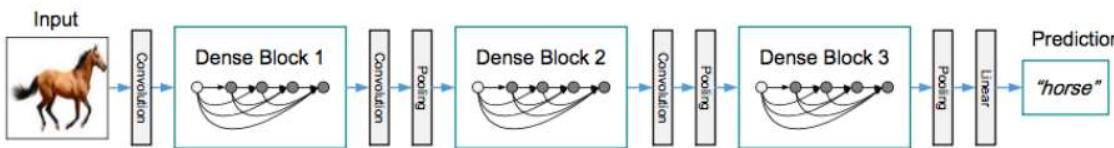


Figure 2: A deep DenseNet with three dense blocks. The layers between two adjacent blocks are referred to as transition layers and change feature-map sizes via convolution and pooling.

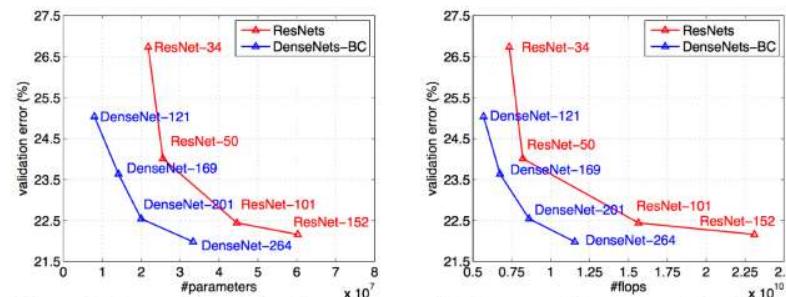


Figure 3: Comparison of the DenseNets and ResNets top-1 error rates (single-crop testing) on the ImageNet validation dataset as a function of learned parameters (left) and FLOPs during test-time (right).

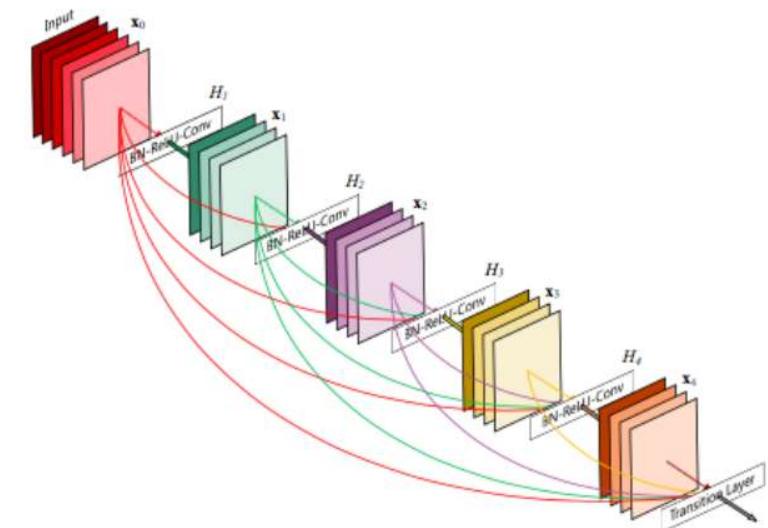


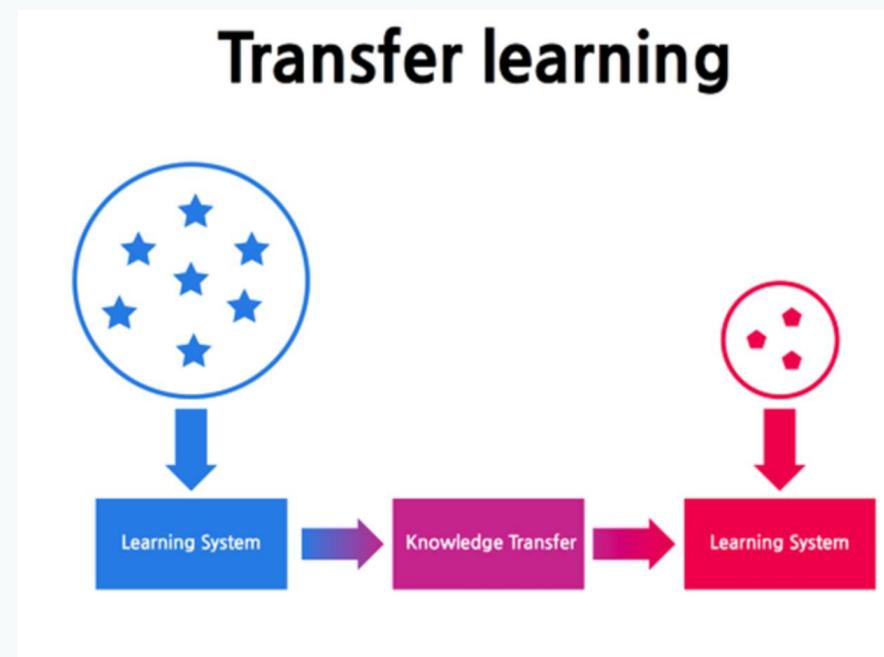
Figure 1: A 5-layer dense block with a growth rate of $k = 4$. Each layer takes all preceding feature-maps as input.

2. Transfer Learning

Transfer Learning with CNNs

Transfer Learning 이란?

기존의 만들어진 모델을 사용하여 새로운 모델을 만들시 학습을 빠르게 하며, 예측을 더 높이는 방법.



2. Transfer Learning

Transfer Learning with CNNs

Transfer Learning 이란?

기존의 만들어진 모델을 사용하여 새로운 모델을 만들시 학습을 빠르게 하며, 예측을 더 높이는 방법.

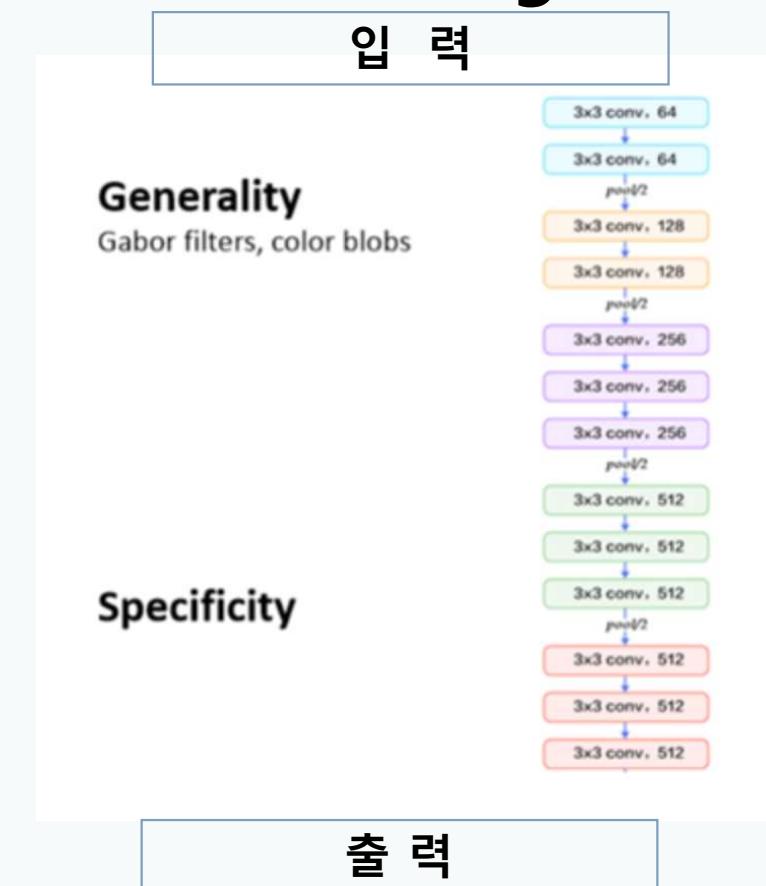
Transfer Learning 이점

- 복잡한 모델일수록 학습시키기 어려운 것을 해결 할 수 있음
- 학습시간이 단축 됨

결론적으로 이미 잘 훈련된 모델이 있는 경우 transfer learning을 적용.
데이터의 도메인에 따라 조금씩 방법이 다름.

2. Transfer Learning

Transfer Learning with CNNs

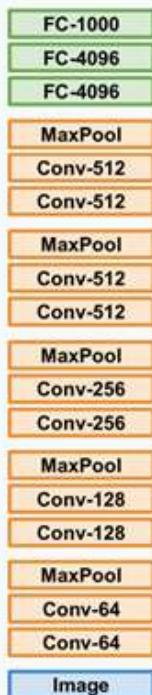


2. Transfer Learning

Transfer Learning with CNNs

Donahue et al, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", ICML 2014
Razavian et al, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition", CVPR Workshops 2014

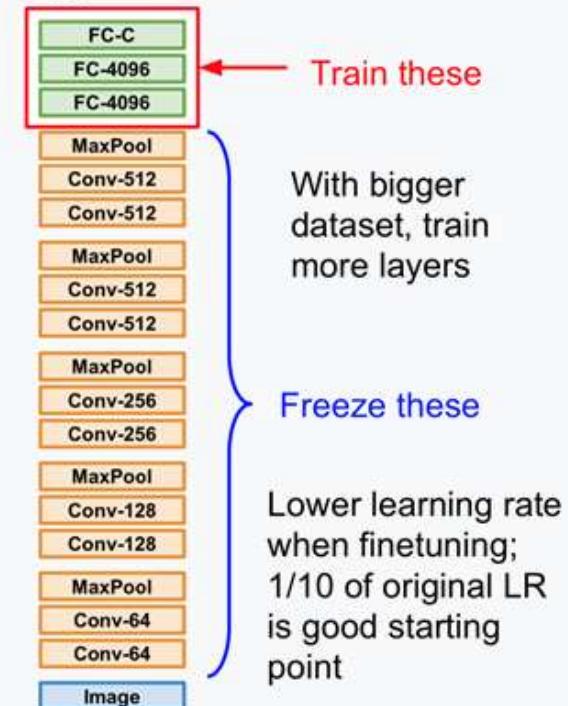
1. Train on Imagenet



2. Small Dataset (C classes)

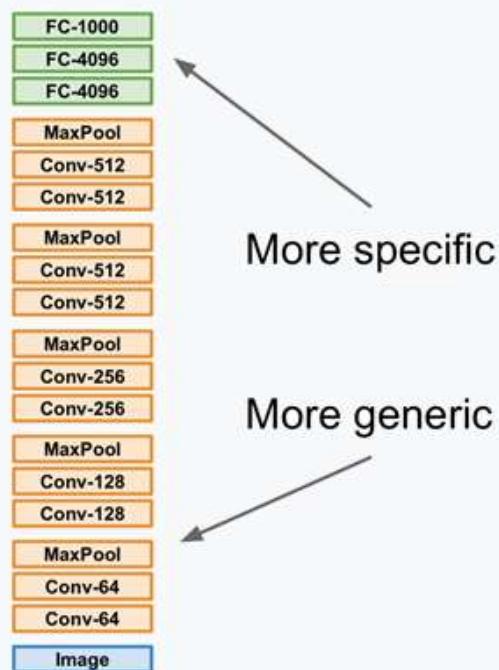


3. Bigger dataset



2. Transfer Learning

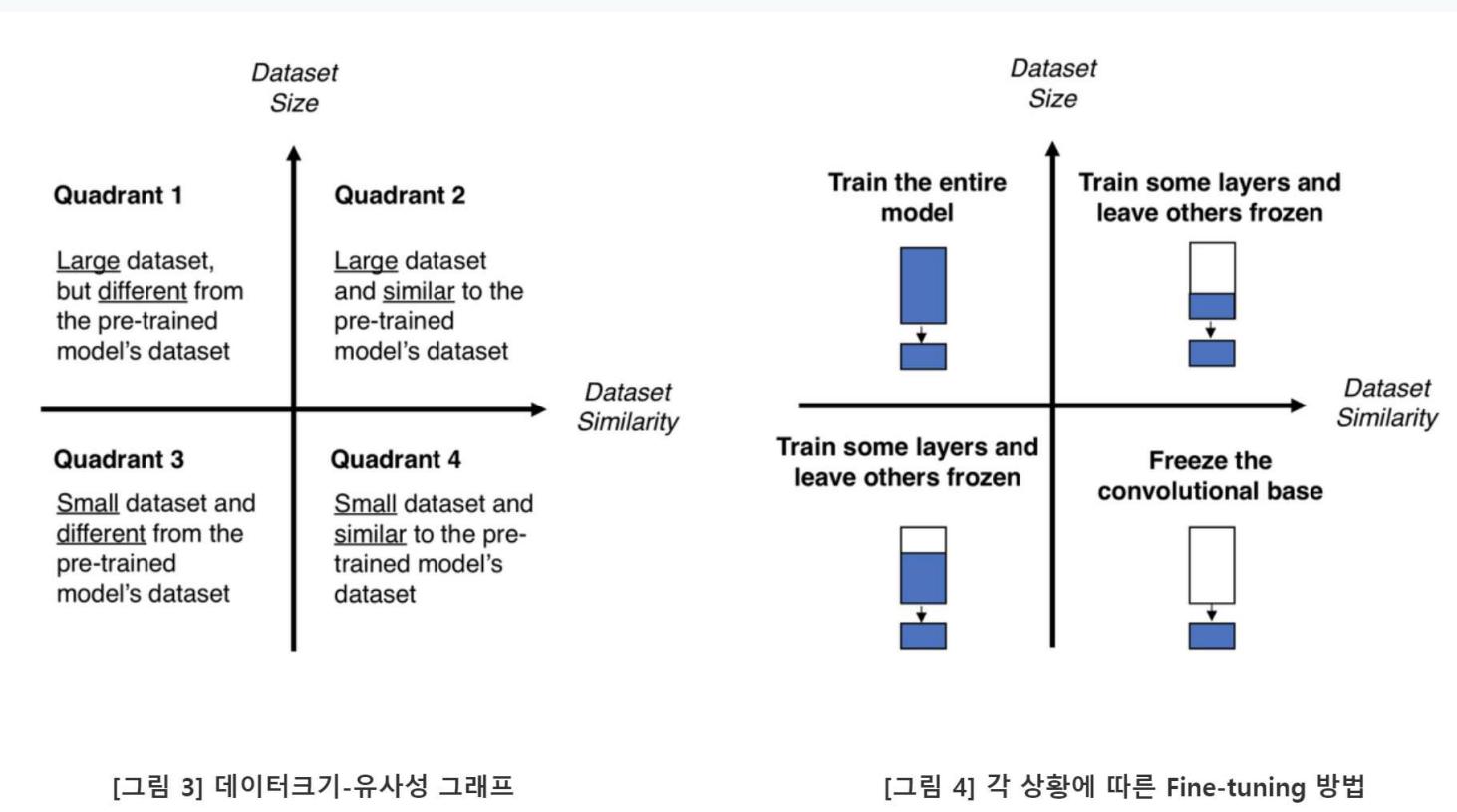
Transfer Learning with CNNs



	very similar dataset	very different dataset
very little data	Use Linear Classifier on top layer	You're in trouble... Try linear classifier from different stages
quite a lot of data	Finetune a few layers	Finetune a larger number of layers

2. Transfer Learning

Transfer Learning with CNNs



[그림 3] 데이터크기-유사성 그래프

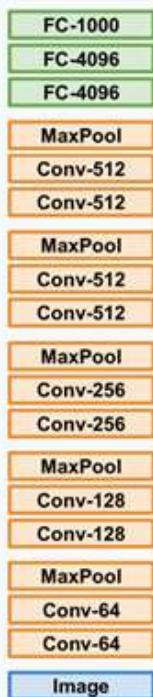
[그림 4] 각 상황에 따른 Fine-tuning 방법

2. Transfer Learning

Transfer Learning with CNNs

Donahue et al, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", ICML 2014
Razavian et al, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition", CVPR Workshops 2014

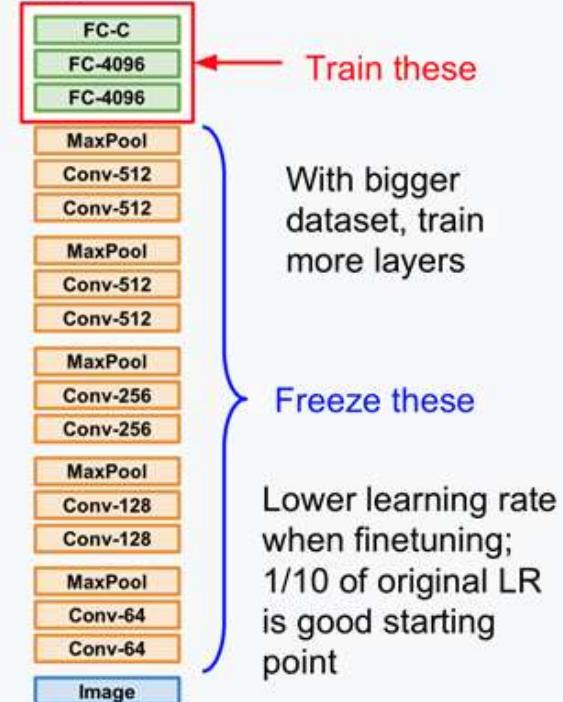
1. Train on Imagenet



2. Small Dataset (C classes)



3. Bigger dataset



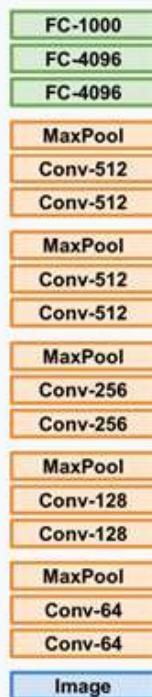
Practice : P_05_00_pretrained_fine_tune.py
944

2. Transfer Learning

Transfer Learning with CNNs

Donahue et al, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", ICML 2014
Razavian et al, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition", CVPR Workshops 2014

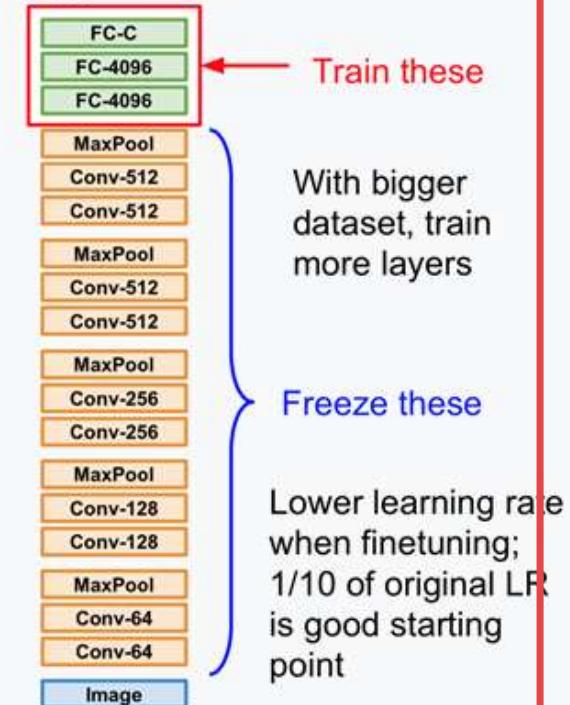
1. Train on Imagenet



2. Small Dataset (C classes)



3. Bigger dataset



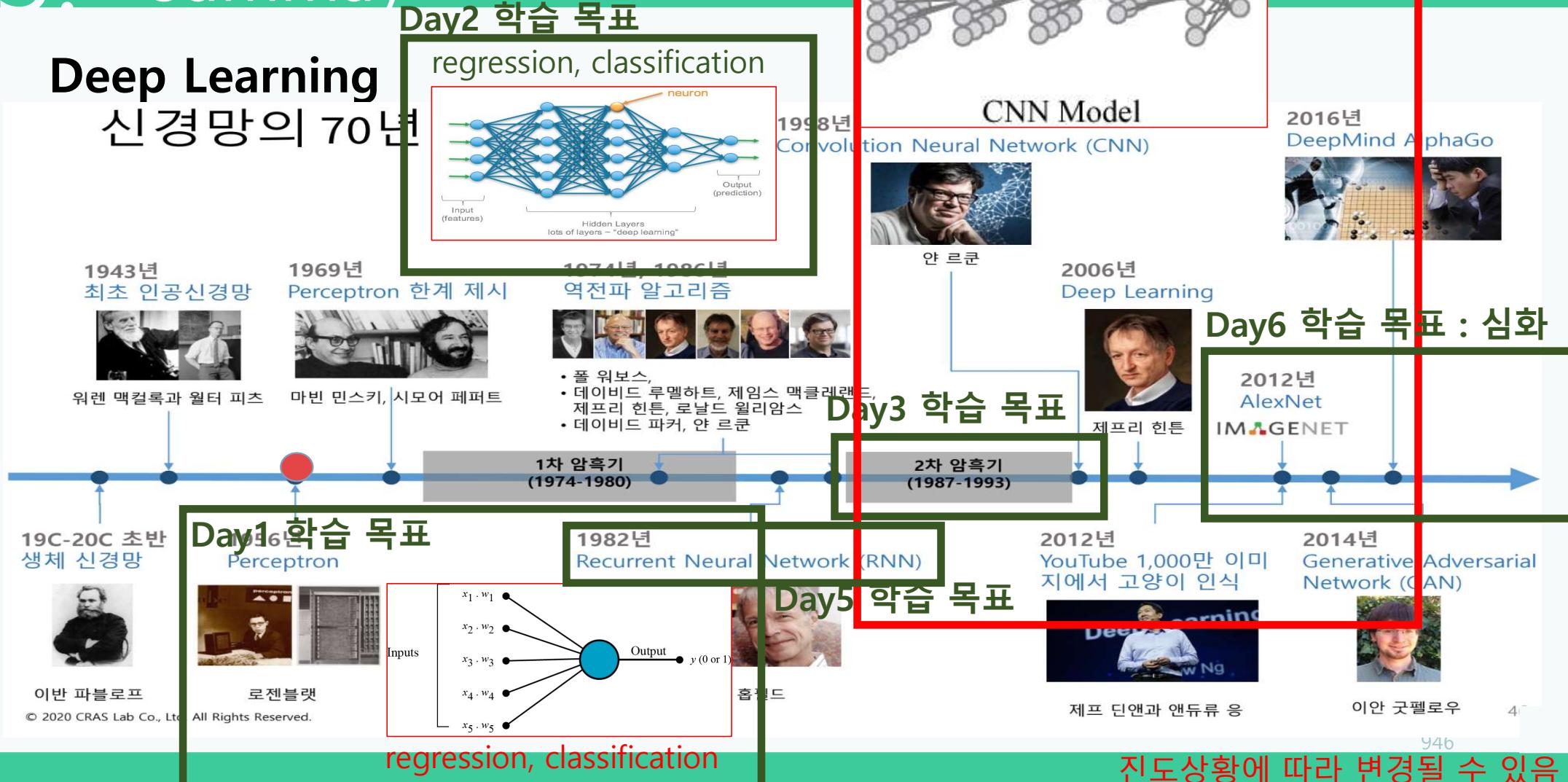
Practice : P_05_01_pretrained_feature_extraction.py

Q45

3. summary

Deep Learning

신경망의 70년



오늘 하루 고생 하셨습니다.
Q & A

참 고 자 료

- cs231n 강의
- 블록과 함께하는 파이썬 딥러닝 케라스 이야기
- 밑바닥부터 시작하는 딥러닝 시즌1,2
- 기타 구글 딥러닝 이미지 및 블로그 자료