



고급 합성곱 신경망

10기 4조 임청수 노지에 한세림



01



Intro

02



LeNet

03



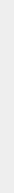
AlexNet

04



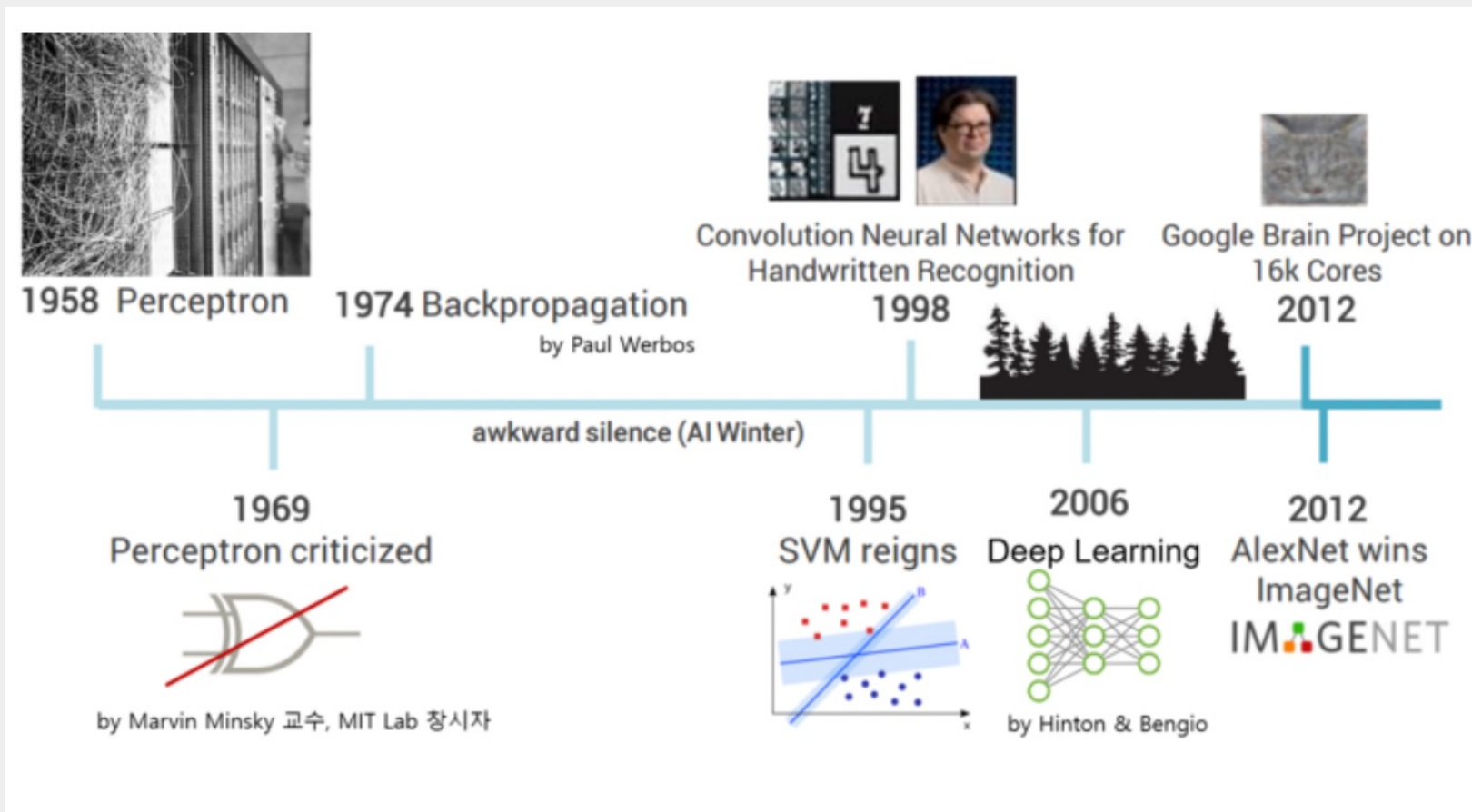
VGGNet

05



ResNet

- ◆ 모델 목표 : 더 깊은 네트워크를 만들면서, 성능을 높여간다.



◆ CNN 모델 발전 흐름

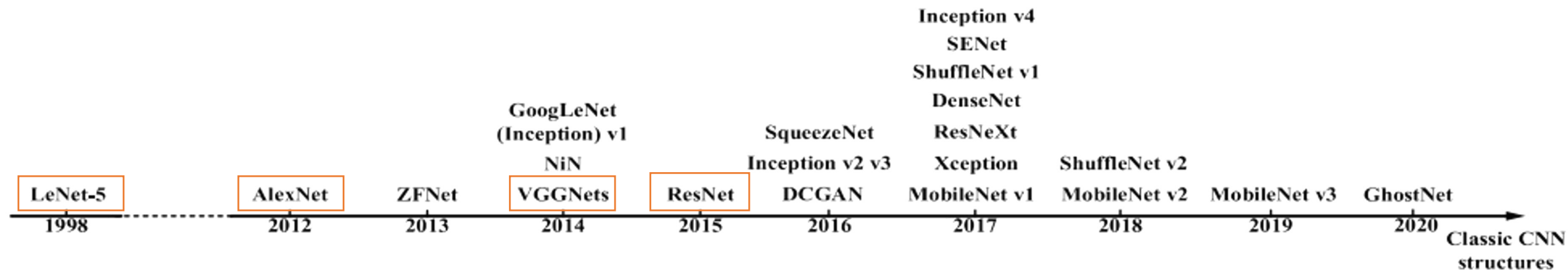
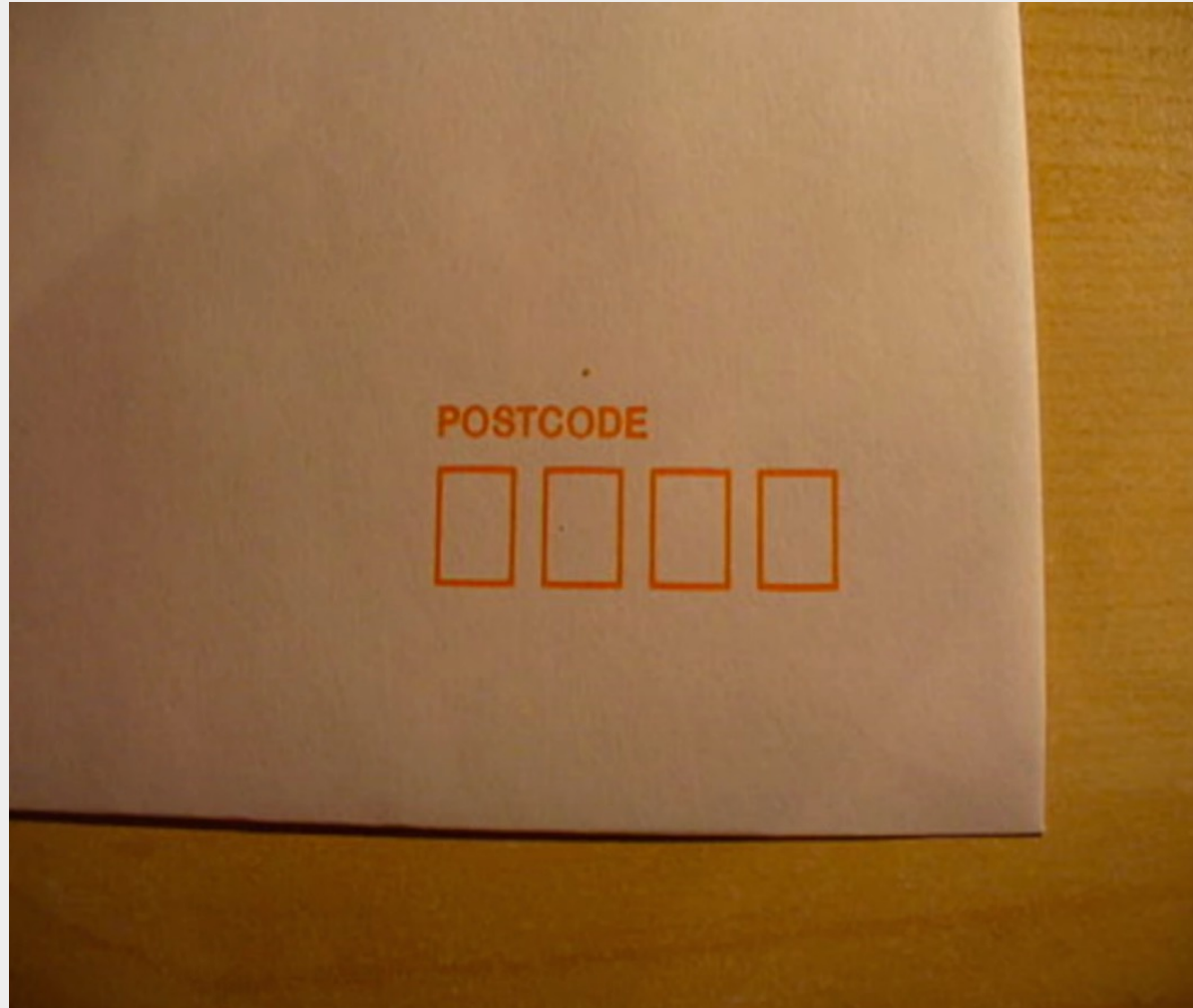


Fig. 4. Part of classic CNN models. NiN: Network in Network; ResNet: Residual Network; DCGAN: Deep Convolutional Generative Adversarial Network; SENet: Squeeze-and-Excitation Network

02 LeNet-5

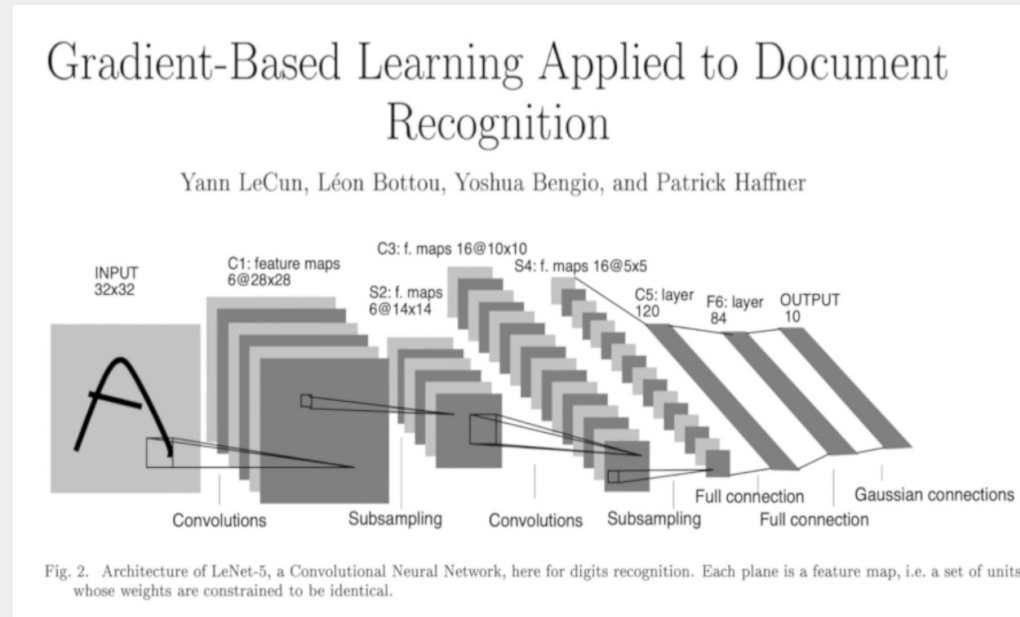




◆ 등장배경

: 이미지 정보 소실 발생과 학습 시간과 효율 문제를 해결하기 위해 1998년 르쿤 연구진이 발표한 모델로, 합성곱 신경망의 선구자로 평가되고 있다.

: 이 모델은 가중치를 가진 5개의 층으로 구성되어 있기 때문에 LeNet-5로 불린다.



- 핵심 아이디어 : Convolution Neural Network 활용
- LeNet 구조 : Input – C1 – Tanh – S2 – C3 – Tanh – S4 – C5 – Tanh – FC6 - softmax

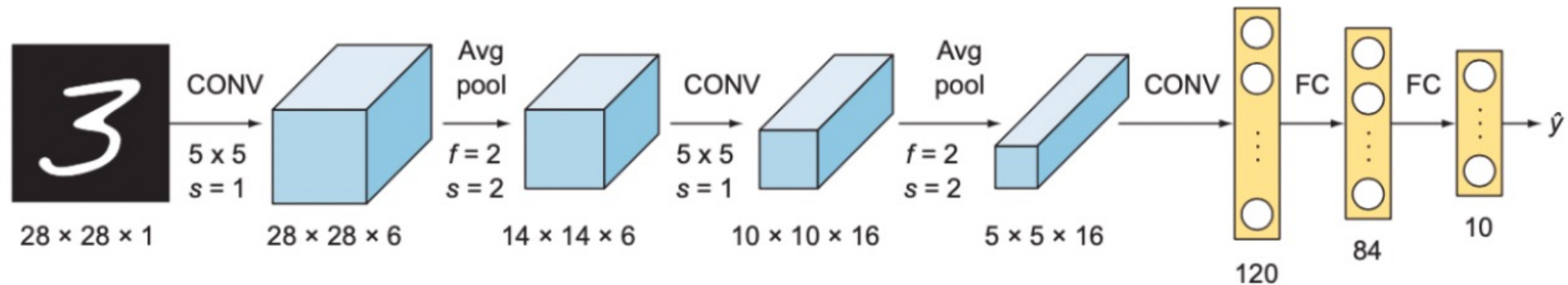


Figure 5.4 The LeNet architecture consists of convolutional kernels of size 5×5 ; pooling layers; an activation function (tanh); and three fully connected layers with 120, 84, and 10 neurons, respectively.

- 모델 특징

1. 네트워크가 깊어질수록 높이와 폭이 줄어들고 채널 수 증가
2. 시그모이드, 당시 ReLU 활성화 함수가 사용되지 않아 tanh 사용
3. 그래프 변환 네트워크를 논문에서 설명하지만 최근에는 잘 쓰이지 않음

- 모델 성능

: MNIST Dataset을 기준으로 99.05% 정확도 달성

- learning rate decay

: LE-Net은 미리 설정된 일정(epoch 수)에 맞춰 학습률을 감소시키는 학습률 감쇠를 활용

```
def lr_schedule(epoch):  
    if epoch <= 2:  
        lr = 5e-4  
    elif epoch > 2 and epoch <= 5:  
        lr = 2e-4  
    elif epoch > 5 and epoch <= 9:  
        lr = 5e-5  
    else:  
        lr = 1e-5  
    return lr
```

←
lr is 0.0005 for the first two epochs, 0.0002 for the next three epochs (3 to 5), 0.00005 for the next four (6 to 9), then 0.00001 thereafter (more than 9).

◆ 등장배경

: 2012년 ILSVRC 이미지 분류 콘테스트에서 우승을 차지한 모델







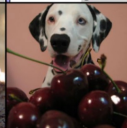
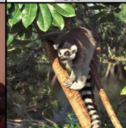
◆ 이미지넷(ImageNet)이란?

: 이미지넷은 시각적 물체 인식 소프트웨어 연구를 위해 설계된 대규모 이미지 데이터베이스이다. 120만장의 이미지, 1,000가지 이상의 클래스로 구성되어 있다.

ImageNet Challenge

IMAGENET

- 1,000 object classes (categories).
- Images:
 - 1.2 M train
 - 100k test.

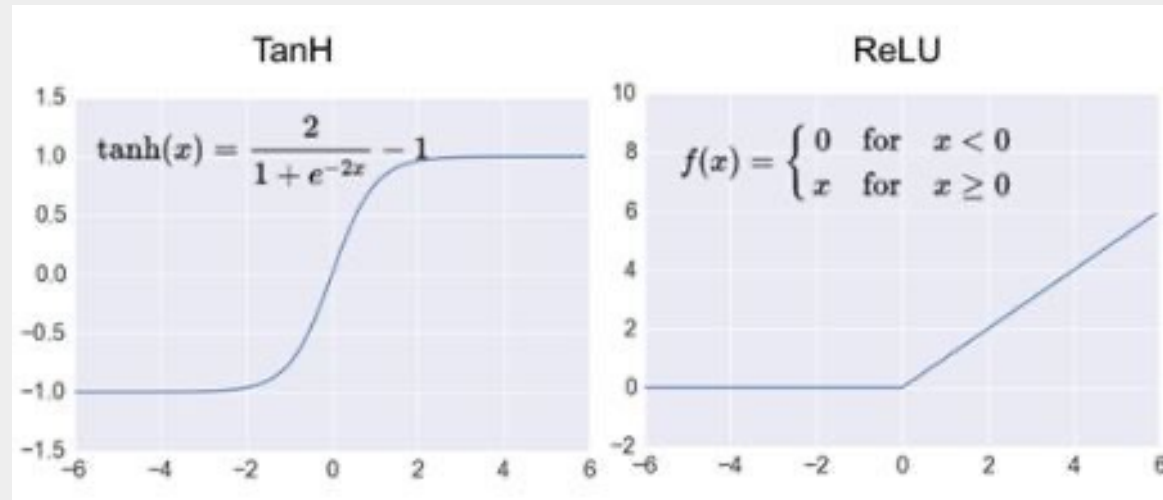
			
mite	container ship	motor scooter	leopard
black widow	lifeboat	go-kart	jaguar
cockroach	amphibian	moped	cheetah
tick	fireboat	bumper car	snow leopard
starfish	drilling platform	golfcart	Egyptian cat
			
grille	mushroom	cherry	Madagascar cat
convertible	agaric	dalmatian	squirrel monkey
pickup	mushroom	grape	spider monkey
beach wagon	jelly fungus	elderberry	titi
fire engine	gill fungus	fordshire bullterrier	indri
	dead-man's-fingers	currant	howler monkey

◆ 등장배경

: LeNet는 복잡한 task(얼굴, 객체 인식 등)가 불가능했다. 따라서 기존 모델의 과적합을 방지하며 학습을 최적화하기 위해 설계되었다.

◆ 핵심 아이디어

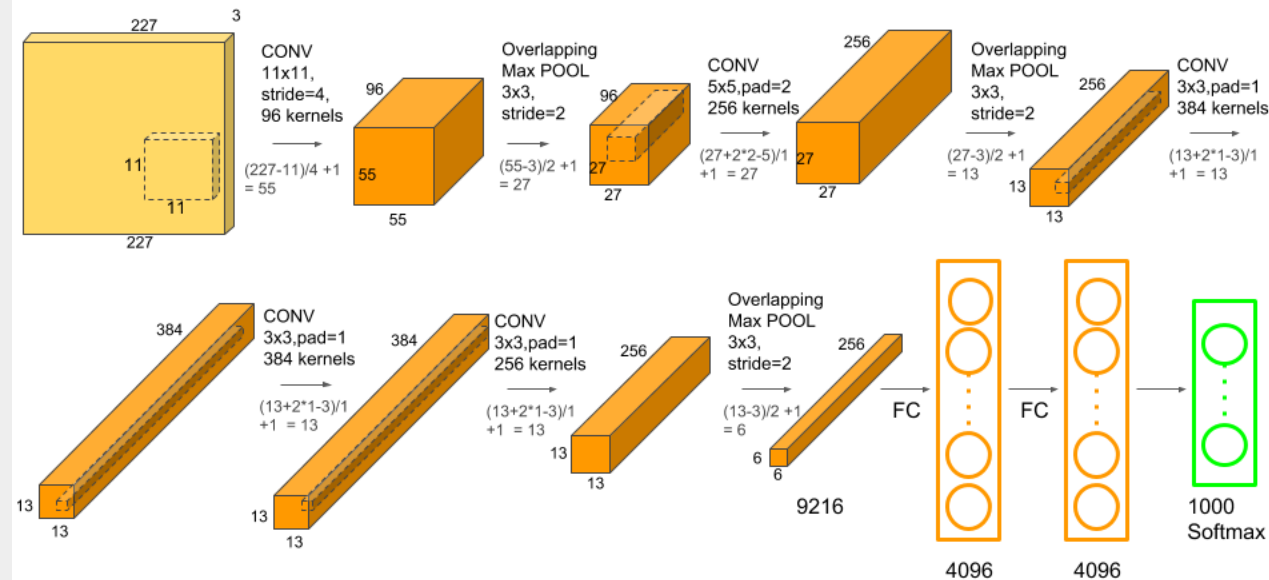
: 활성화함수 ReLU, dropout, Data Augmentation 등



◆ 모델 구조

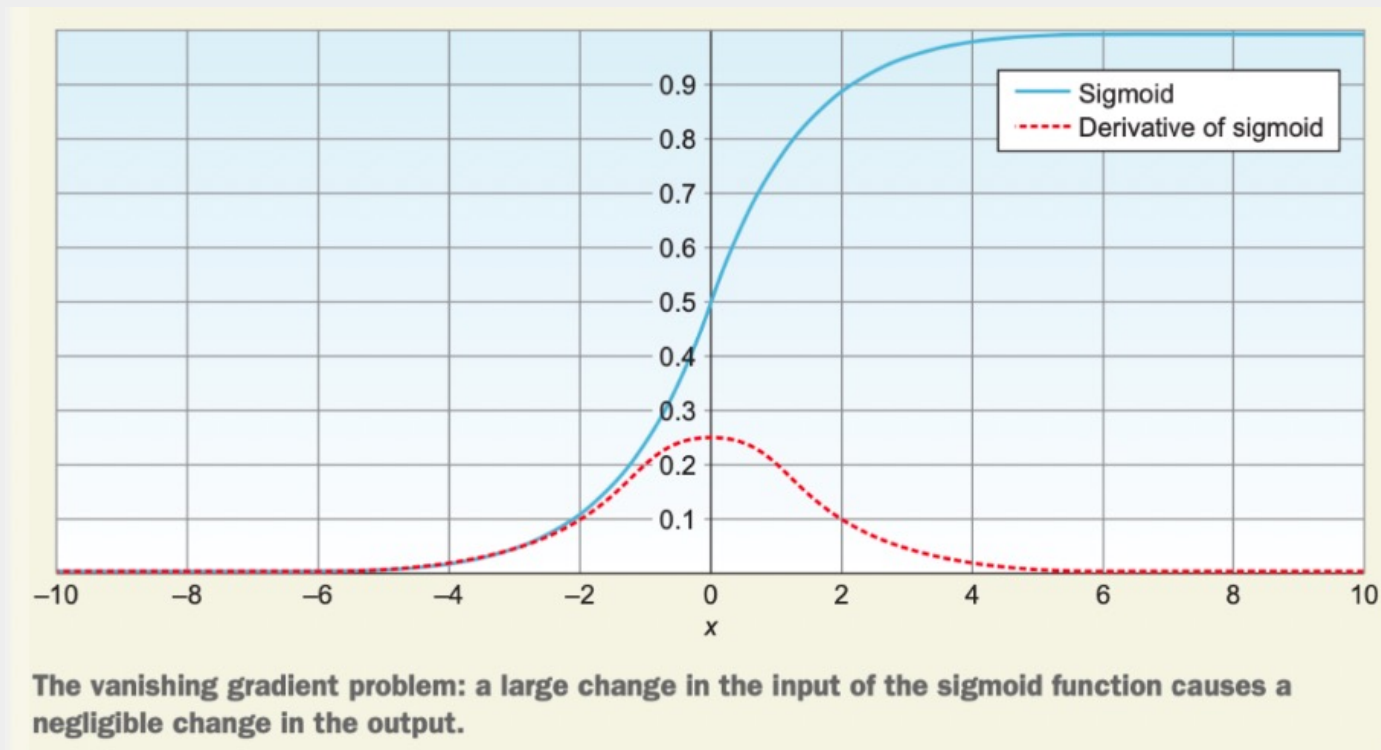
: CONV1 – POOL2 – CONV3 – POOL4 – CONV4 – CONV5 – CONV6 – CONV7 – POOL8 – FC9 – FC10 – SOFTMAX7

: 필터의 크기 : 11x11, 5x5, 3x3(다양한 특성을 추출하기 위함)



◆ AlexNet의 개선사항

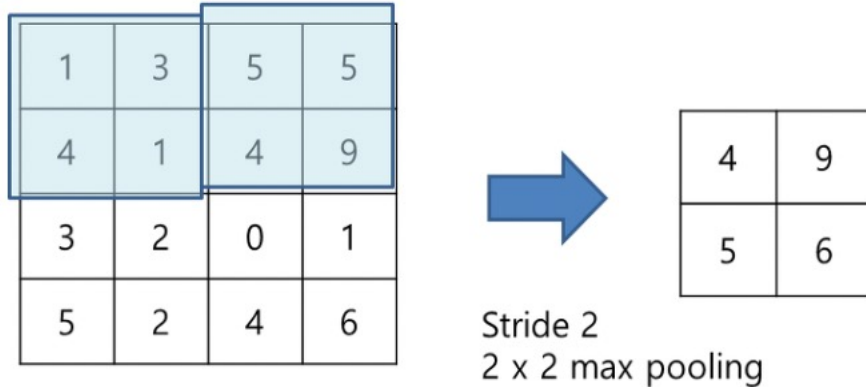
: AlexNet은 신경망에 비선형성을 도입하기 위해 tanh나 시그모이드 대신 ReLU 함수를 사용하면서 기울기 소실 문제를 해결하였다.



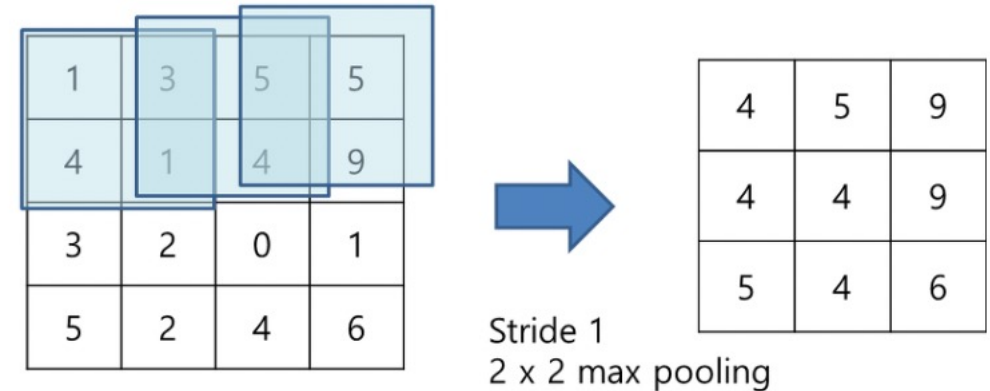
◆ AlexNet의 개선사항

: max pooling을 사용하였다. 그리고 overlapping pooling을 통해 풀링르 겹치는 방식으로 사용하였다.

Non-overlapping pooling



Overlapping pooling



◆ 드롭아웃

개념 : 특정 뉴런을 비활성화하여 순방향 계산과 역전파 계산에서 모두 배제한다.

효과 : 이를 통해 뉴런 간의 상호 적응을 방지(특정 뉴런에 overfitting)하며 다양한 조합의 뉴런에 도움을 주는 유용한 특징 학습 가능하다. AlexNet에는 두 FC layer에 모두 0.5비율의 드롭아웃이 적용되었다.

◆ 데이터 증강

: AlexNet에는 두 가지 증강 기법이 사용되었다. 하나는 이미지 변환과 수평반사이며 다른 하나는 훈련 이미지에 서 RGB 채널의 강도를 변경하는 것이다.

◆ 국소 응답 정규화(Local Response Normalization, LRN)

: CONV나 POOL 레이어에서 작업을 진행할 때 매우 높은 하나의 픽셀값이 주변의 픽셀에 영향을 미치게 되므로, 이런 부분을 방지하기 위해 다른 특징맵의 같은 위치에 있는 픽셀끼리 정규화하는 방법이다.

◆ 모델 성능

: 2012 ILSVRC에서 15.3%의 top-5 오차율을 기록했으며, 2위 모델 오차율은 26.2%로 큰 격차를 보였다

top-1 오차율과 top-5 오차율

두 오차율은 알고리즘의 분류 성능을 나타내기 위한 개념입니다.

top-1 오차율은 분류기가 정답 클래스에 가장 높은 확률을 부여하지 않은 비율입니다.

top-5 오차율은 정답이 예측 확률 상위 5개 안에 들어 있지 않은 비율입니다.

즉 top-5 오차율은 예측 결과가 틀렸더라도 정답과 얼마나 근접했는지 나타내는 지표입니다.

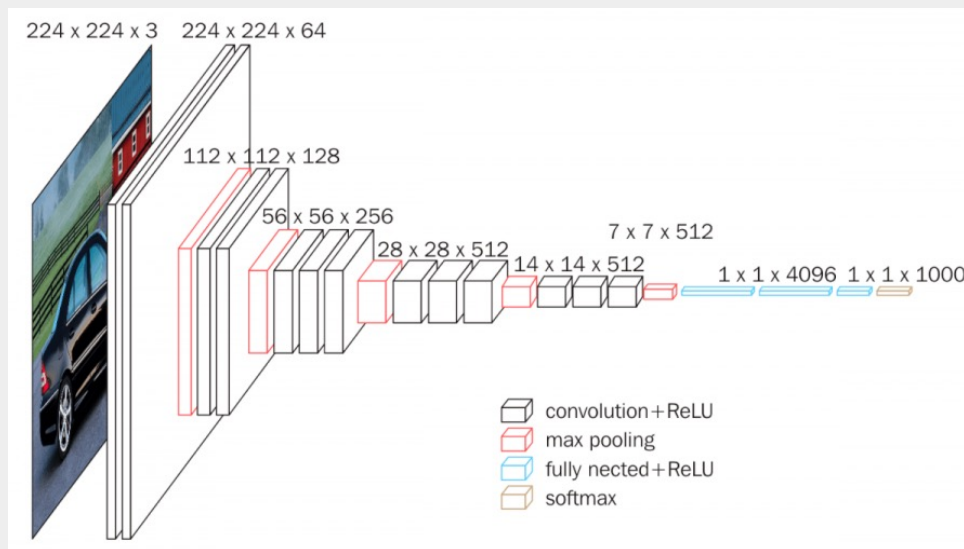
◆ 등장배경

: AlexNet이 나온 후 성능을 높이기 위해 깊은 네트워크를 쌓는 연구를 진행하였다.

◆ 핵심 아이디어

: filter size를 3x3으로 고정하면서 깊은 네트워크를 형성하였다.

: 층이 쌓이면서 높이, 너비는 작아지는데 필터는 2배로 계속 규칙적인 상승을 보인다.



◆ AlexNet과 비교

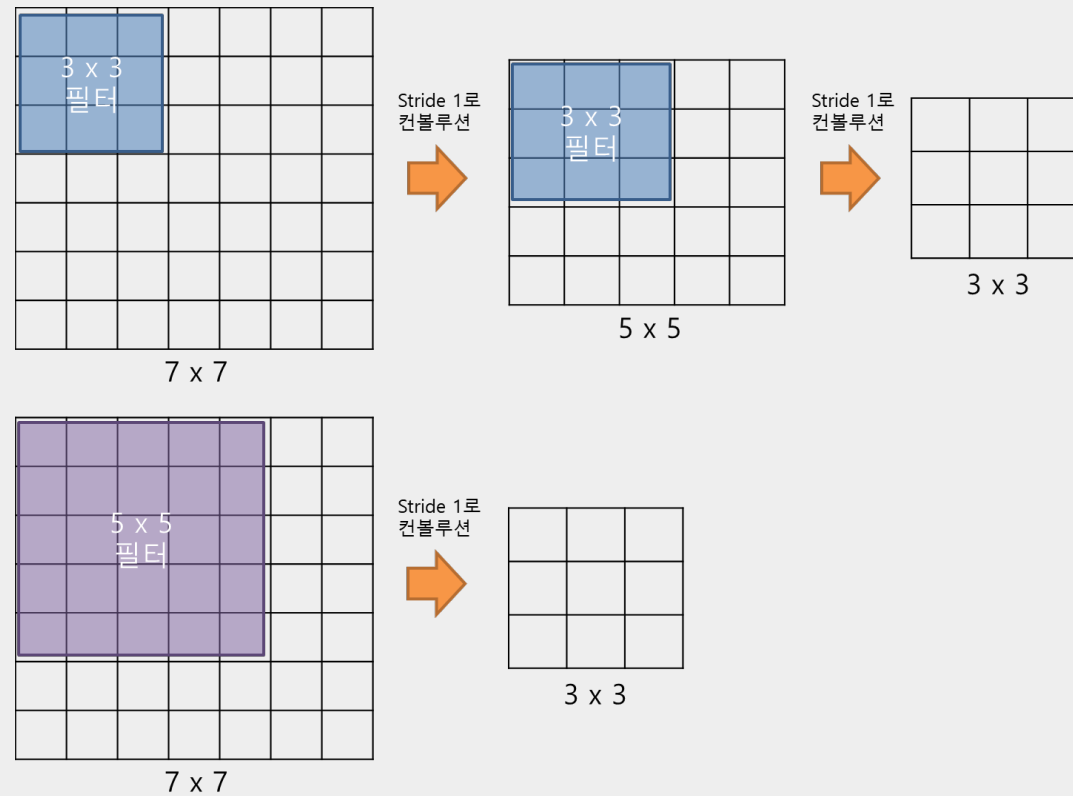
1. VGGNet의 개선점은 동일하게 설정된 층(합성곱층과 FC층)을 사용해서 신경망 구조를 단순화시켰다는 것이다.
2. VGGNet의 신경망 구조는 일련의 합성곱층 뒤에 풀링층이 배치되는 구조이며 합성곱층과 풀링층은 다음과 같다.

: 모든 합성곱층은 3x3 크기의 필터 사용, 스트라이드와 패딩은 각각 1,1 적용

: 모든 풀링층은 2x2 크기의 풀링 영역을 사용, 스트라이드는 2 적용

- VGGNet에서 합성곱층의 필터 크기를 줄이면서 더 세밀한 특징을 추출할 수 있음
- 수용 영역의 크기가 같을 때 크기가 큰 하나의 커널보다 크기가 작은 커널을 여러 개 쌓은 쪽이 더 성능이 좋음
- 이는 커널을 여러 개 쌓으면서 커널 뒤에 비선형층(ReLU)를 거치는데 신경망 층수를 늘리는 것과 동일한 효과가 있음

◆ Layer는 다르지만 수용영역이 동일한 예



◆ 모델 구조

: 3x3 크기의 커널을 여러 층 쌓은 합성곱층 사이에 2x2 크기의 풀링층을 넣는 식으로 구성

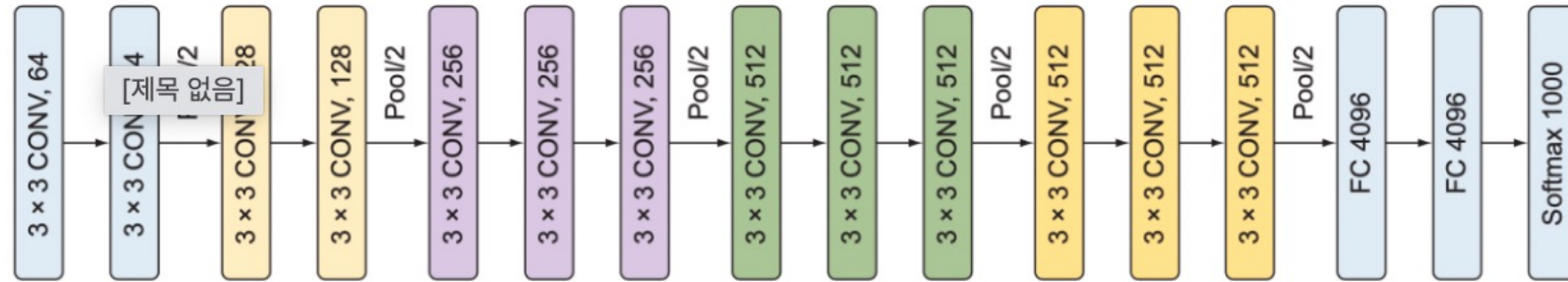
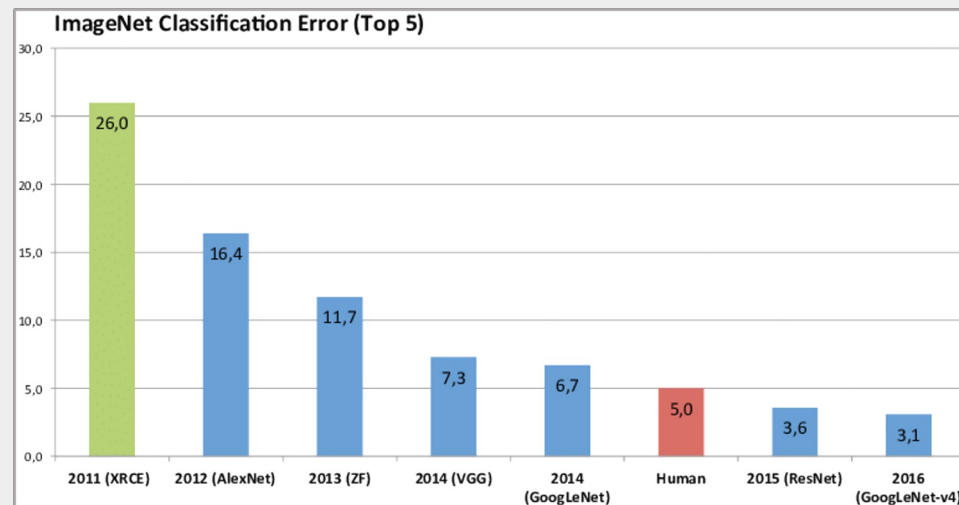


Figure 5.8 VGGNet-16 architecture

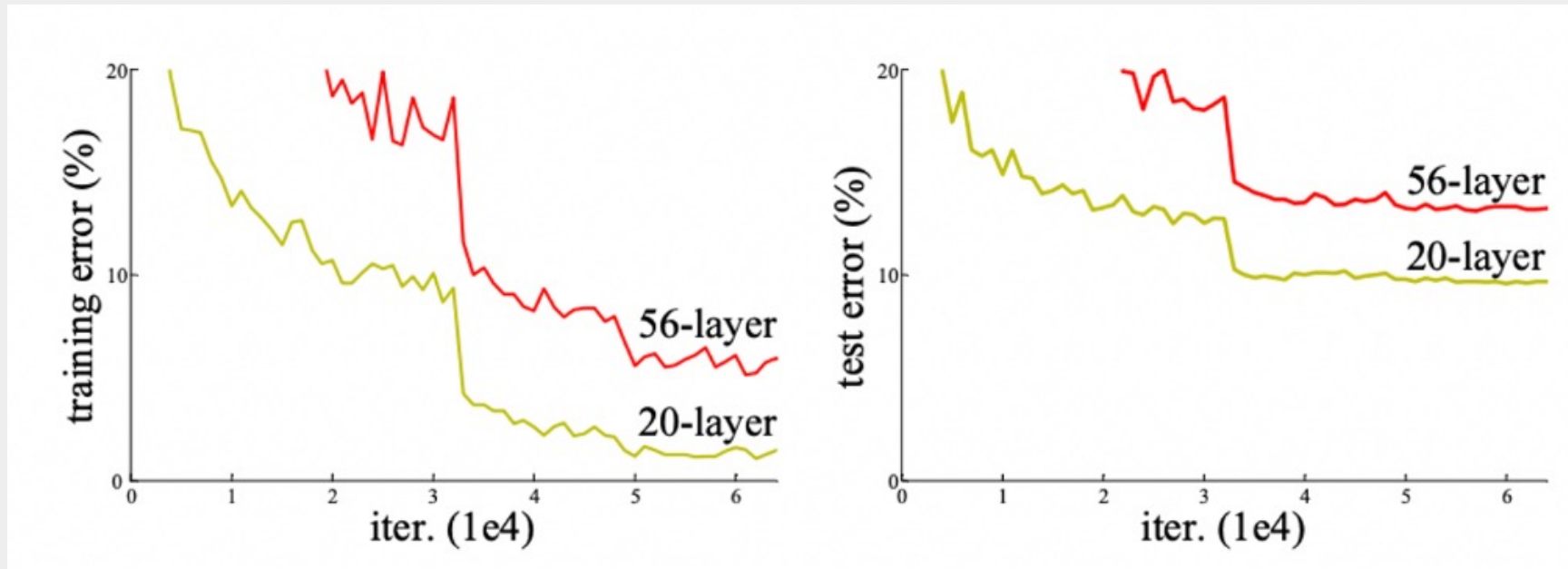
◆ 등장배경

- ResNet은 2015년에 마이크로소프트 리서치 팀에서 제안한 신경망 구조이다.
- 잔차모듈과 스킵 연결이라는 새로운 구조가 사용되었으며, 은닉층에도 강한 배치 정규화가 사용되었다. 이를 통해 50층, 101층, 152층이나 되는 깊은 신경망임에도 불구하고 VGGNet19보다 더 복잡도가 낮음을 보여주었다.
- 2015년 ILSVRC에서 top-5 오차율 3.57%을 기록하며 1위를 했다.



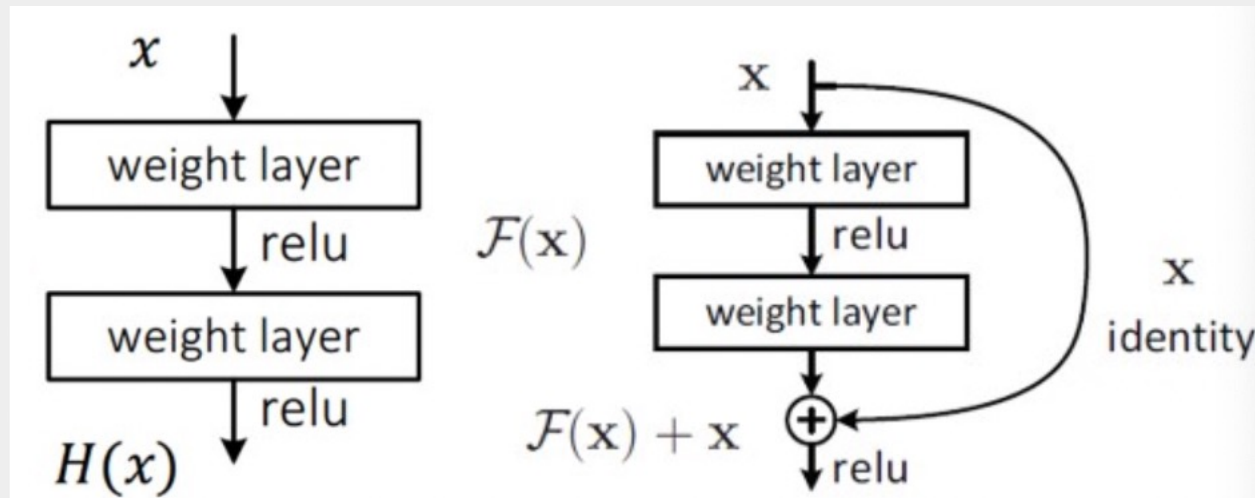
◆ Degradation

- layer가 깊어지면서 train과 test가 학습되지 않는 현상
- 두 데이터셋 모두 error가 증가하므로 overfitting과는 다르다



◆ Residual learning

- $H(x)$ 가 아닌 출력과 입력의 차이인 $H(x)-x$ 를 얻도록 목표를 수정. 따라서 Residual Function인 $F(x) = H(x)-x$ 를 최소화해야 함
- 즉 입력값과 출력값의 차이가 0이 되도록 해야 하며 이는 $H(x)-x = 0$, $H(x) = x$ 인 identity mapping(항등 함수)를 의미함
- 그 결과, 레이어는 Direct로 학습하는 것 대신에, skip connection을 통해 각각의 layer들이 작은 정보들을 추가적으로 학습하도록 함(각각의 레이어가 배워야 할 정보량을 축소)



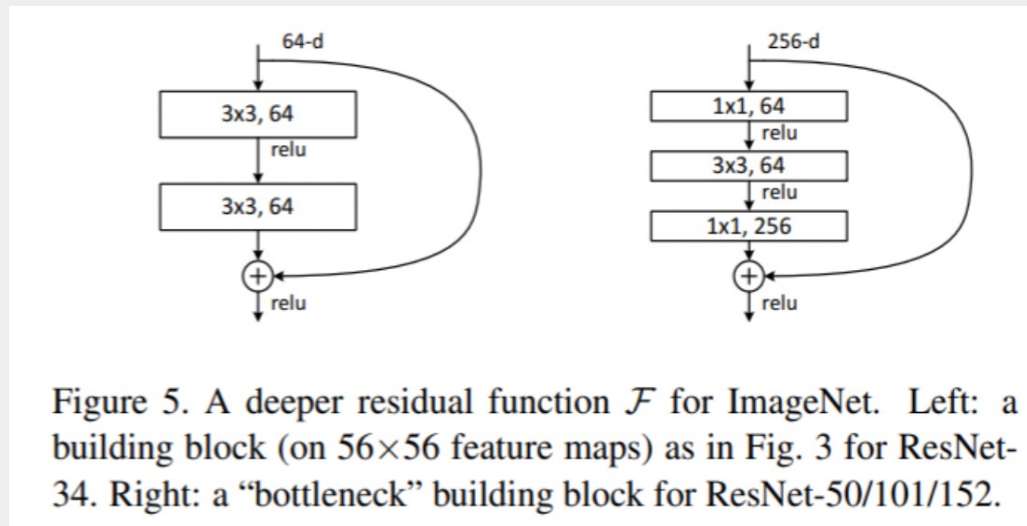
◆ Bottleneck

: 네트워크가 깊어짐에 따라 증가하는 복잡도를 줄이기 위해 적용
차원 축소를 통해 파라미터 수를 줄일 수 있지만 정보 손실 발생

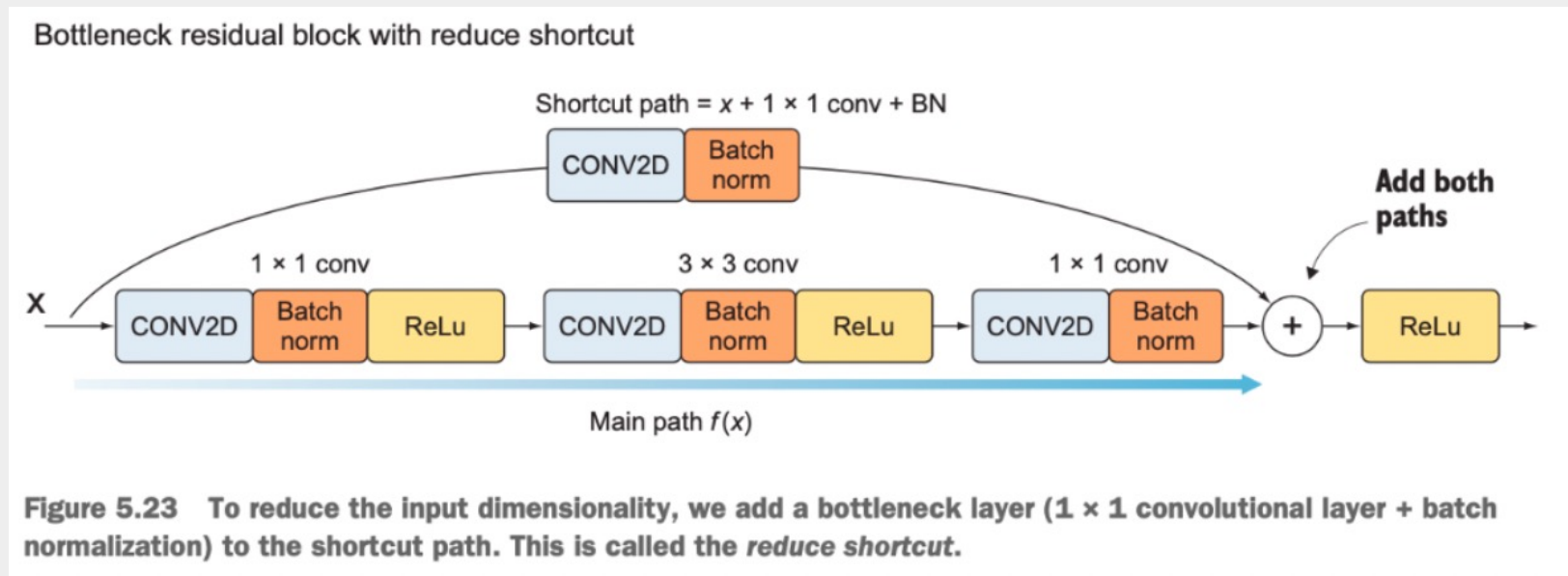
◆ 연산량 식 : $\text{input channel} \times \text{output channel} \times (\text{kernel size})^2 \rightarrow \text{sum}(\text{layer})$

Standard : $64 \times 64 \times 3 \times 3 \times 2 = 73728$

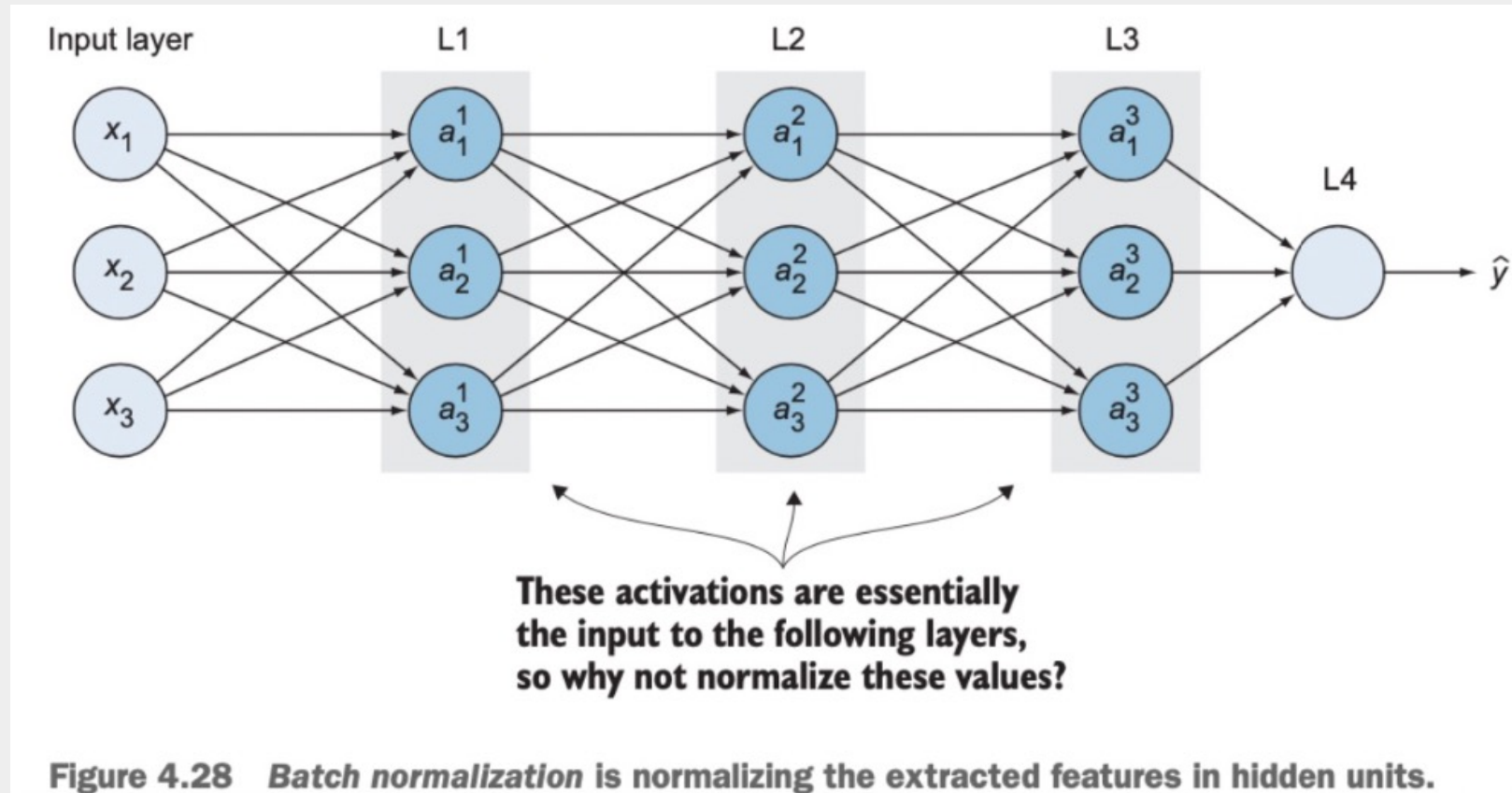
For resnet-50/101/152 : $16384 + 36864 + 16384 = 69632$



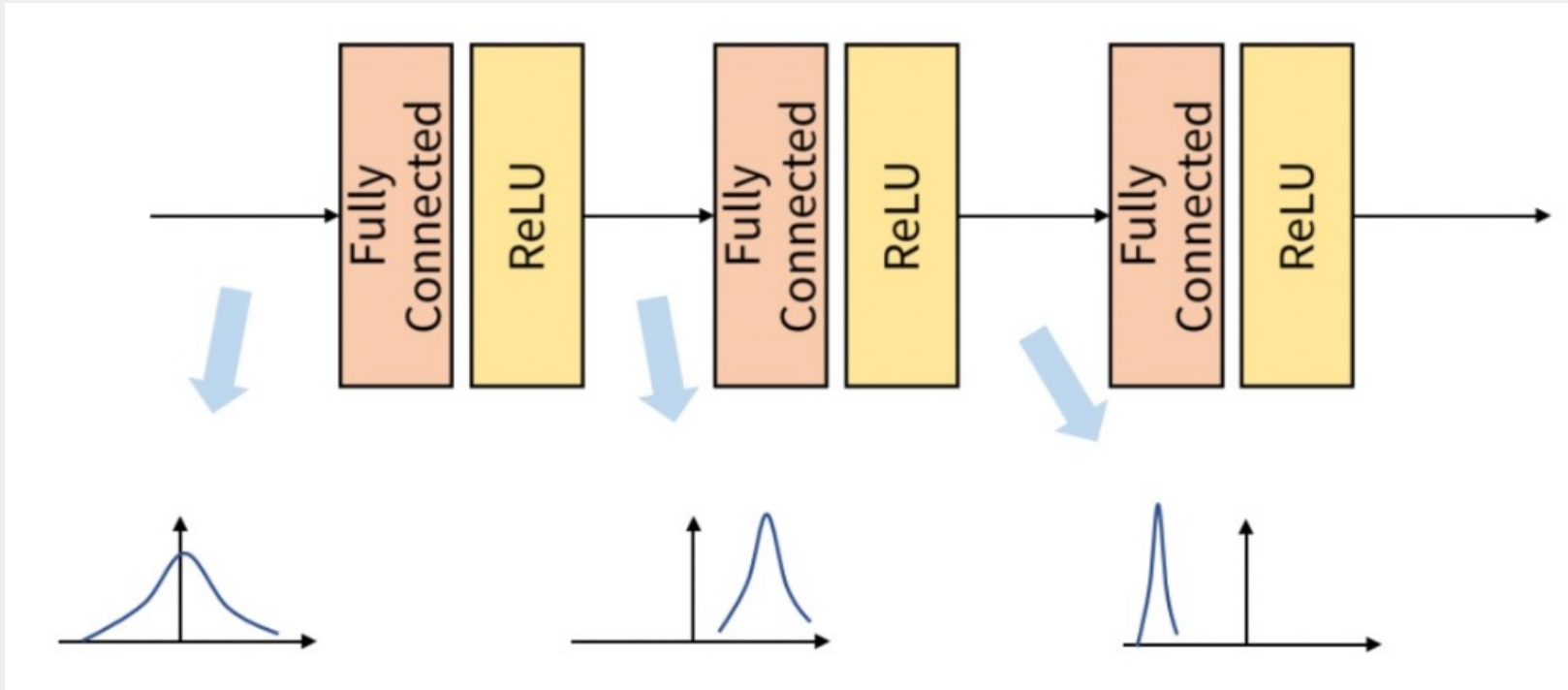
- ◆ **Batch Normalization** : 은닉층 뉴런의 출력이 항상 표준 분포를 따르도록 강제하는 방법



- ◆ **Batch Normalization** : 일반적으로 정규화는 입력층에 집중되는데, 각 은닉층을 통해 추출된 특징을 정규화를 통해 vanishing / exploding gradient 문제를 해결함

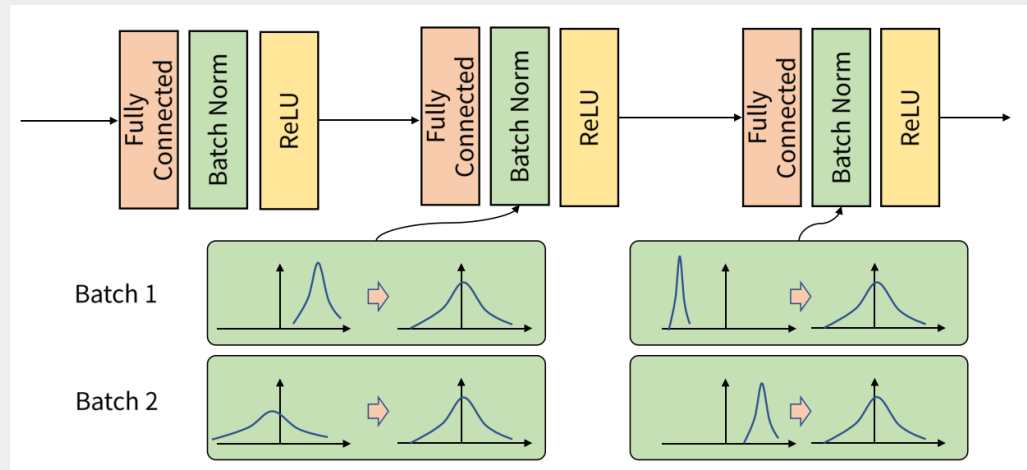


- ◆ **Internal covariate shift** : 학습과정에서 이전 레이어의 파라미터 변화로 인해 현재 레이어 입력의 분포가 바뀌는 현상. 레이어가 깊어질수록 심해진다.



◆ BN 설명

- 학습을 batch 단위로 진행하기 때문에 정규화도 mini batch 단위로 진행된다.
- Batch Normalization은 각 배치 별 평균을 빼고 표준편차로 나누어준다. 따라서 데이터의 분포는 평균은 0, 표준편차는 1로 조절할 수 있다.
- 일반 정규화와 차이점은 scale 과 shift 연산을 위한 γ 와 β 가 있다는 것이다. γ 와 β 는 데이터를 정규화할 때 활성화 함수가 비선형성을 잃는 것을 완화한다.



✨ ✨ 배치 정규화의 수학적 원리

1. 입력의 평균을 0으로 조정하기 전에 입력의 평균과 표준편차를 계산해야 합니다. m 이 배치 내 데이터 수일 때 미니배치의 평균 μ_B 과 표준편차 σ_B 는 다음과 같이 계산합니다.

$$\mu_B \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad (1)$$

$$\sigma_B^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad (2)$$

✨ ✨ 배치 정규화의 수학적 원리

2. 입력을 정규화합니다. 여기서 \hat{x} 은 평균이 0이며 정규화된 입력입니다. 여기서 ε 은 분모가 0이 되는 것을 방지하기 위한 아주 작은 수(10^{-5} 정도)로 설정합니다.

$$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}$$

3. 배율 및 위치를 조정합니다. 정규화된 입력에 γ 를 곱해 배율을 조정하고 β 를 더해 위치를 조정한 y_i 를 계산합니다. 이 두 가지 값은 새로운 파라미터로서 학습 중 최적화할 대상입니다.

$$y_i \leftarrow \gamma X_i + \beta$$



감사합니다
