

표준화 standardization

$$\frac{X - \mu}{\sigma}$$

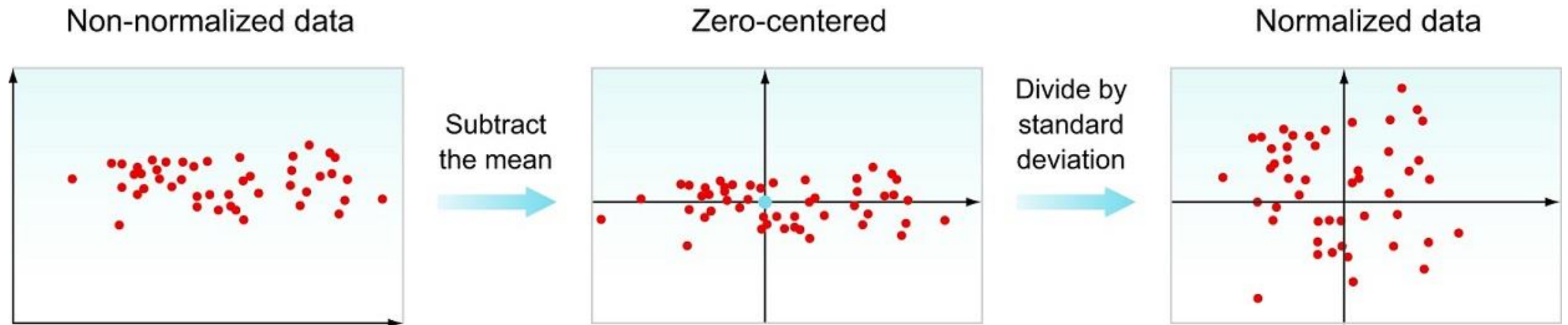
X : 값

μ : 평균

σ : 표준편차

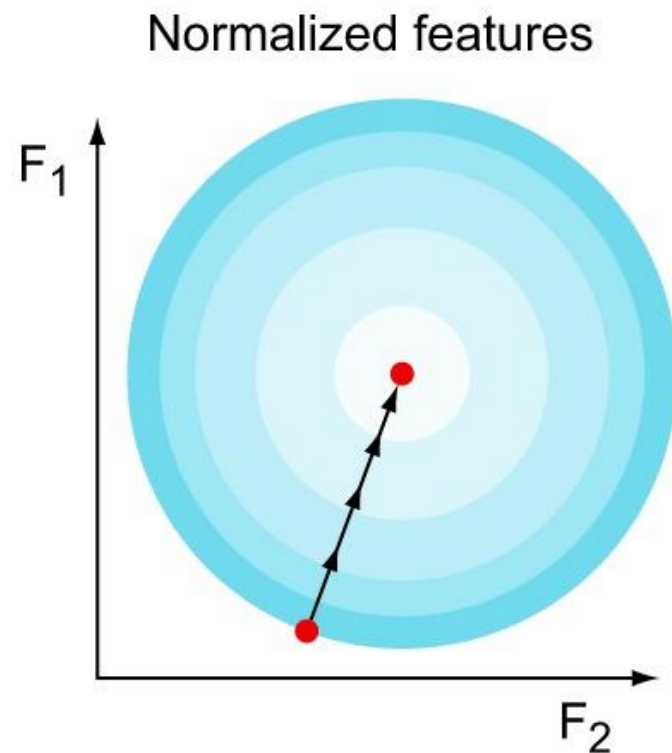
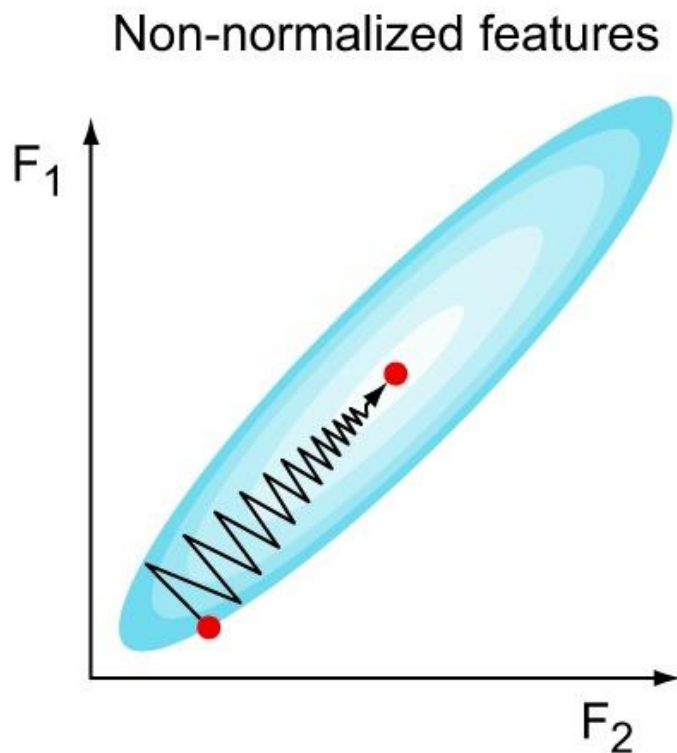
- 평균 = 0, 표준편차 = 1이 됨
- 서로 다른 평균과 표준편차를 가진 변수를 비교하기 쉽게 해줌

Image Normalization



Normalization의 효과

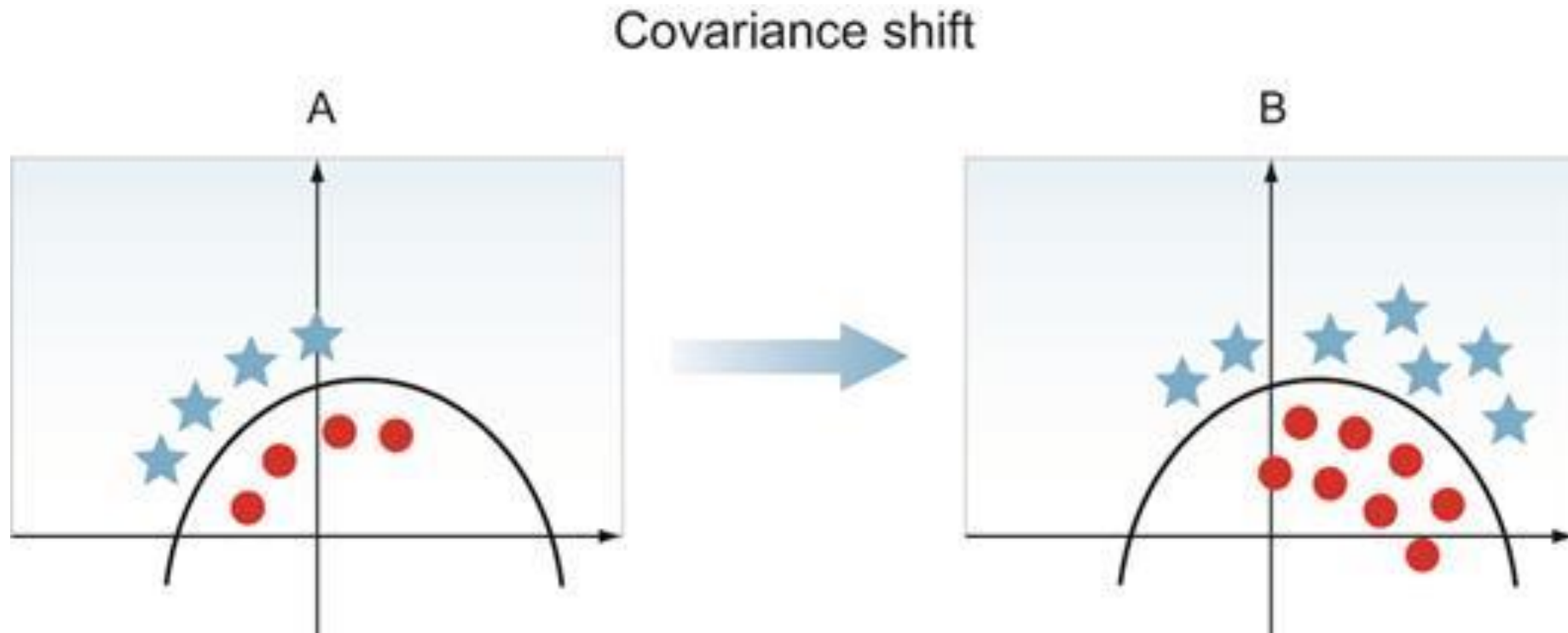
Gradient descent with and without feature scaling



Covariance Shift Problem

- 인공신경망에서 한 레이어의 출력은 다음 레이어의 입력
- 한 레이어는 이전의 입력을 바탕으로 학습
- 한 레이어에서 학습이 이뤄지면 출력이 변화 → 다음 레이어의 입력이 변화
- 입력이 계속 변화하므로 학습이 잘 이뤄지지 않게 됨

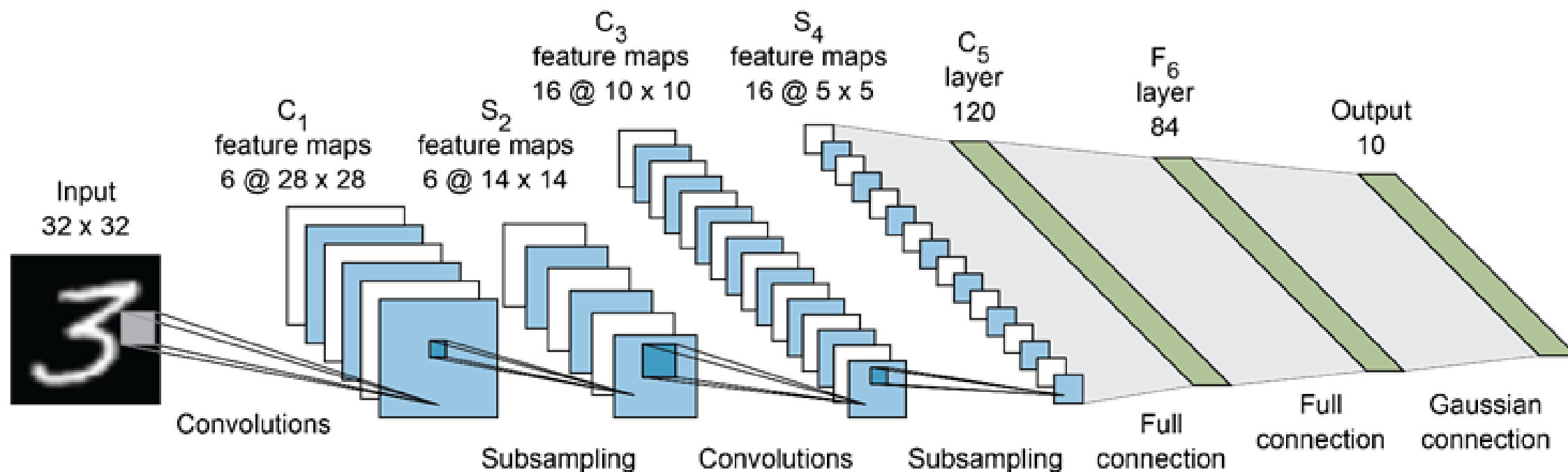
Covariance Shift Problem



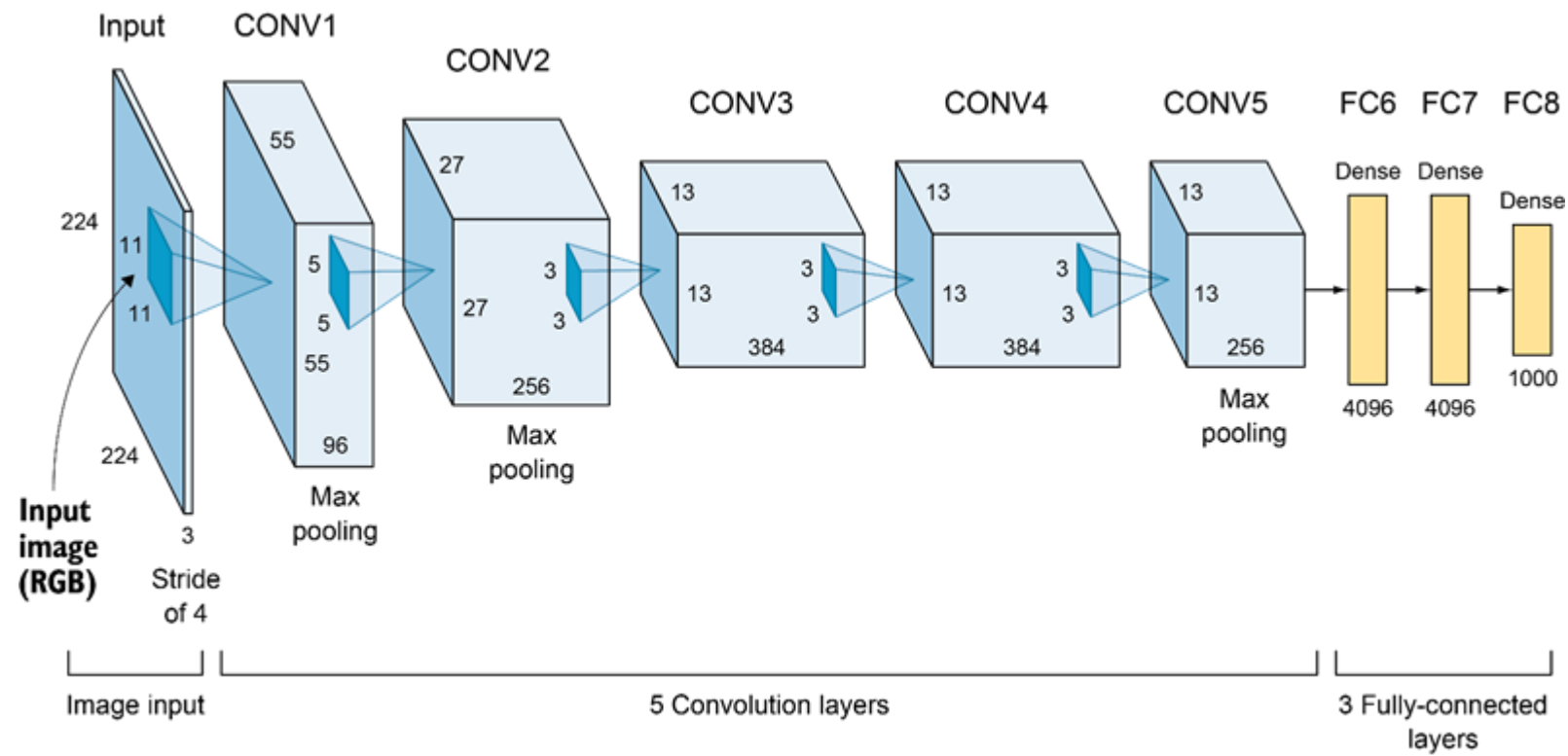
Batch Normalization

- 모형 입력을 normalize 하는 것과 비슷하게 각 레이어의 입력도 normalize하면 성능이 향상될 수 있음
- 모형의 입력과 달리 각 레이어의 입력은 학습 과정에서 매번 해줘야 하므로 계산 비용이 높음
- batch 단위로 normalize 하여 효율을 향상

LeNet-5



AlexNet



AlexNet의 새로운 점

- ReLU
- Dropout
- Batch Normalization과 같은 개념이 추가
- Weight regularization

노름 Norm

- 길이 또는 크기를 일반화한 개념
- Lp 노름

$$\left(\sum_{i=1}^N |w_i|^p \right)^{1/p}$$

L1 정규화

- 손실함수에 파라미터의 L1 노름을 추가

$$J(w) + \alpha \sum_{i=1}^N |w_i|^1$$

- LASSO
- 가중치를 0으로 만드는 경향이 강함

L2 정규화

- 손실함수에 파라미터의 L2 노름을 추가

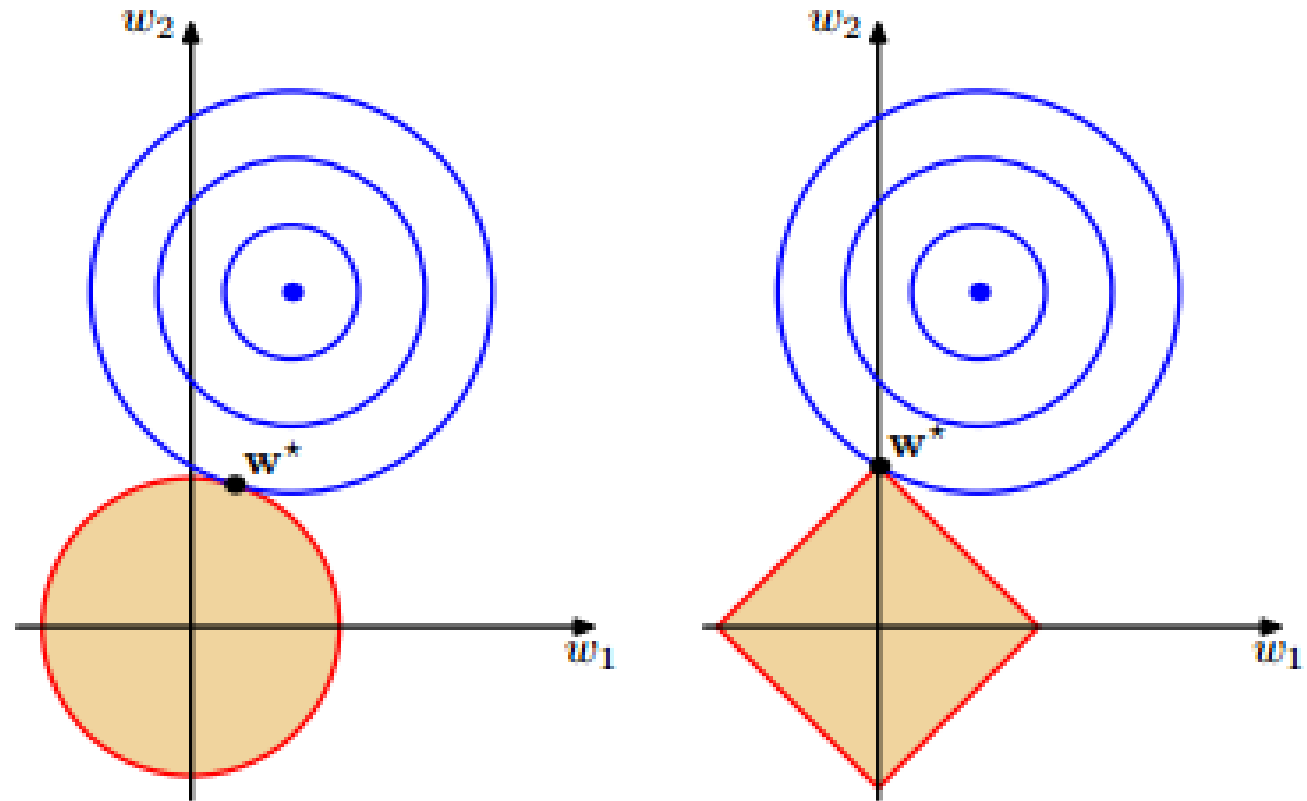
$$J(w) + \alpha \sum_{i=1}^N |w_i|^2$$

- Ridge 회귀분석 or Weight Decay
- 가중치를 전반적으로 작게 만드는 경향이 강함

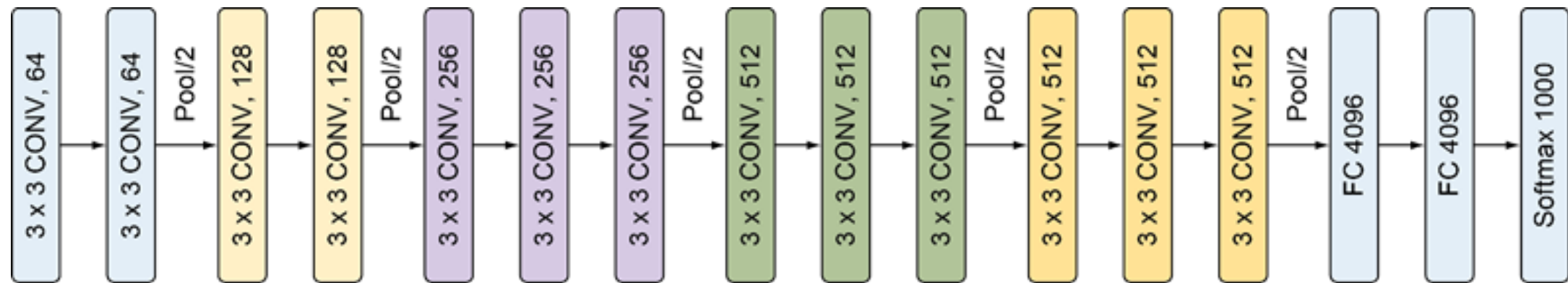
L2 vs. L1

146 3. LINEAR MODELS FOR REGRESSION

Figure 3.4 Plot of the contours of the unregularized error function (blue) along with the constraint region (3.30) for the quadratic regularizer $q = 2$ on the left and the lasso regularizer $q = 1$ on the right, in which the optimum value for the parameter vector w is denoted by w^* . The lasso gives a sparse solution in which $w_1^* = 0$.



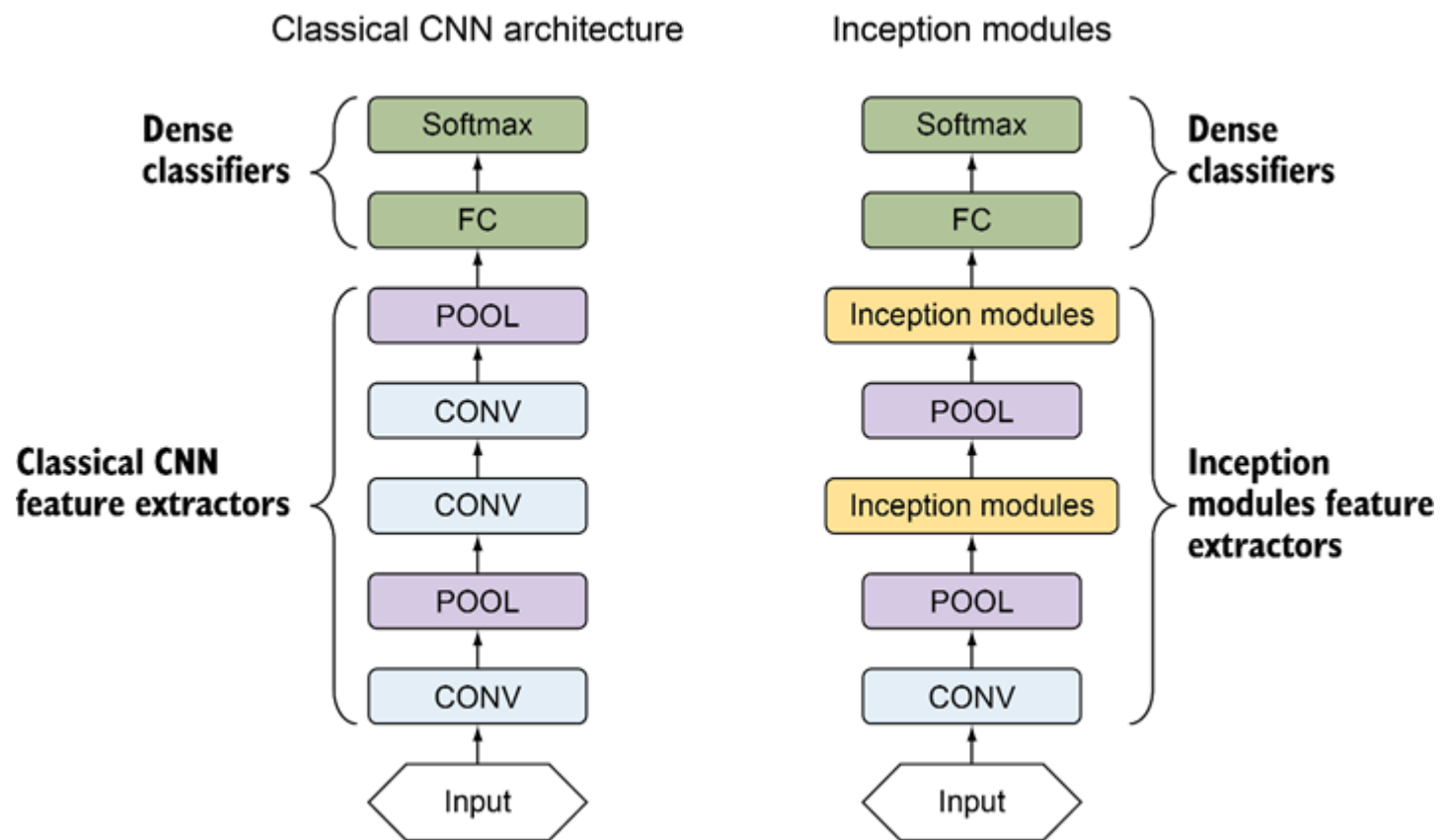
VGGNet



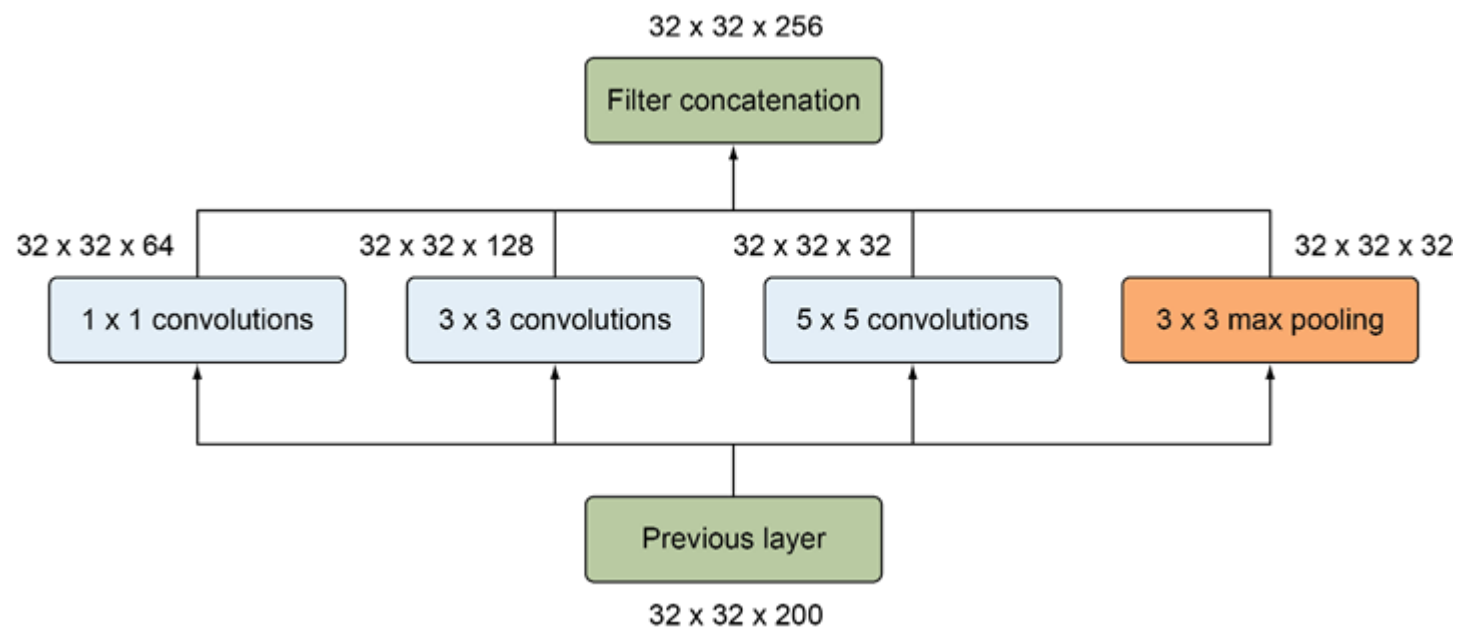
VGGNet

- 합성곱 레이어는 모두 3x3 크기
- 풀링 레이어는 모두 2x2 크기
- 3x3 합성곱 2개는 5x5 합성곱과 receptive field 크기가 같으나 파라미터의 수가 더 적고, 1개의 레이어 대신 2개의 레이어를 사용하므로 비선형성을 추가할 수 있음

GoogLeNet

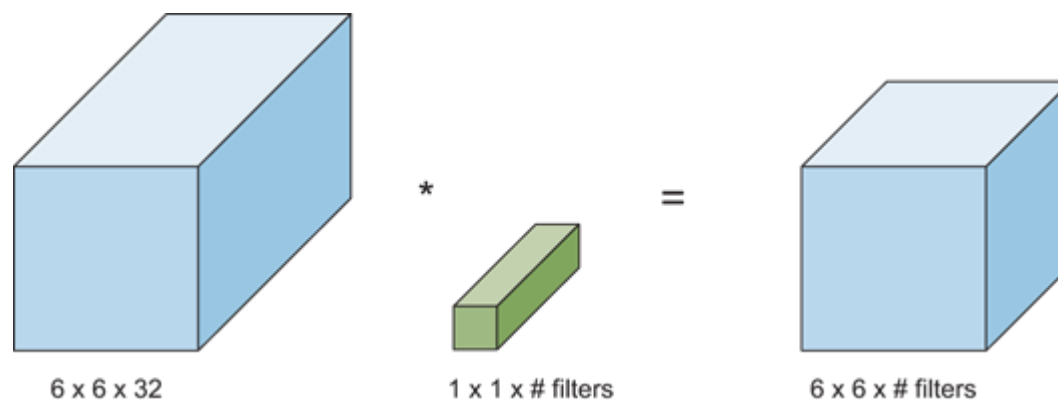


Inception 모듈



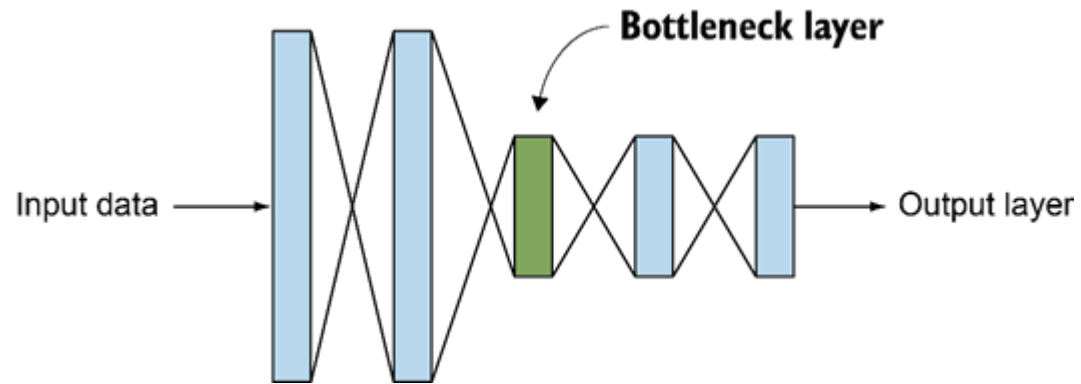
1x1 Conv. Layer

- 채널의 수를 줄이는 효과



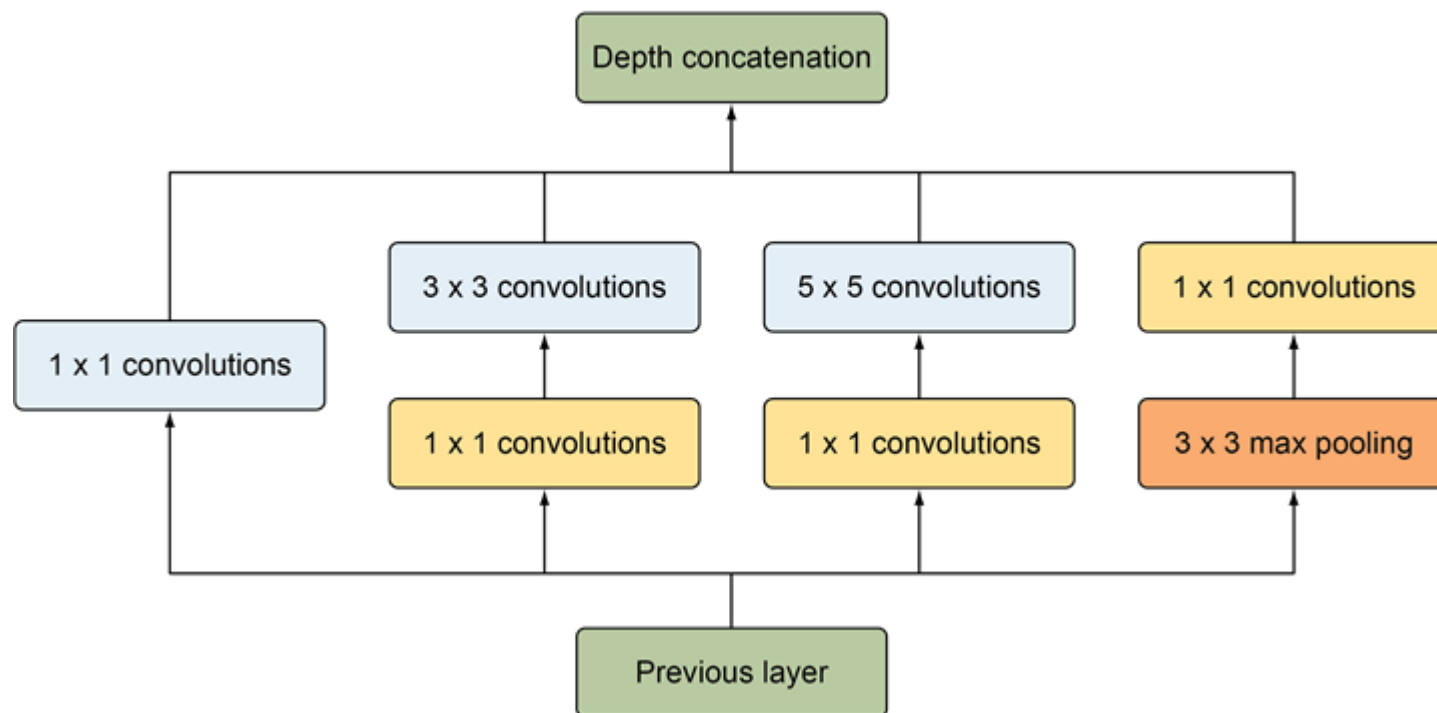
Bottleneck Layer

- 1x1 합성곱 레이어를 통해 채널의 크기를 줄여 계산량을 감소시킴



Inception 모듈 with 차원 축소

Inception module with dimensionality reduction

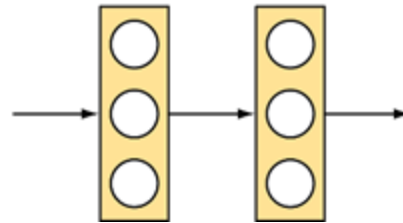


사라지는 경사

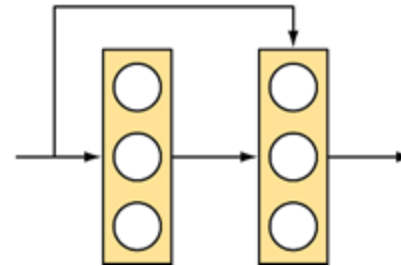
- 매우 깊은 신경망을 만들 경우 초반부의 레이어들의 경사가 매우 작아지거나(사라지는 경사), 또는 매우 커지는(폭발하는 경사) 문제 → 학습이 잘 안 됨

skip connection

Without skip
connection



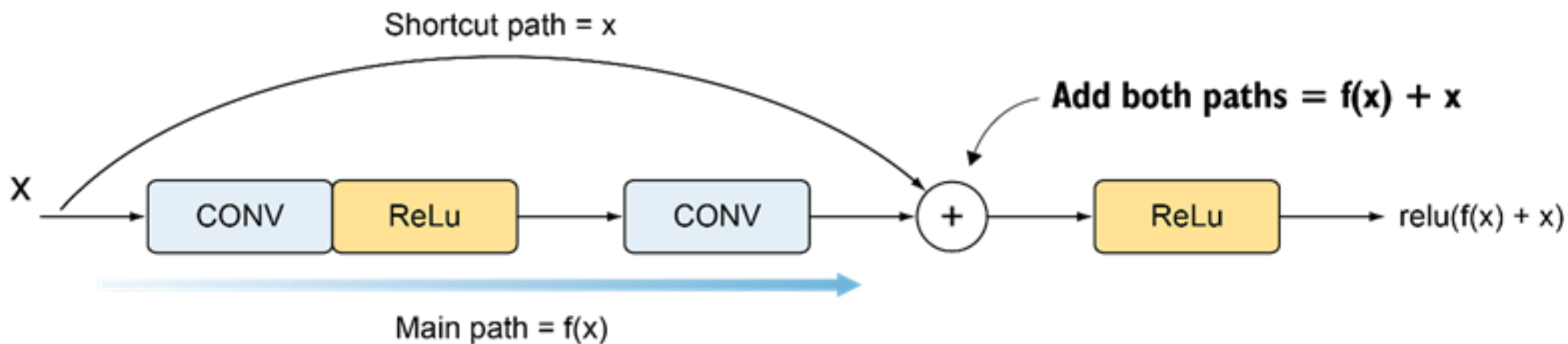
With skip
connection



skip connection

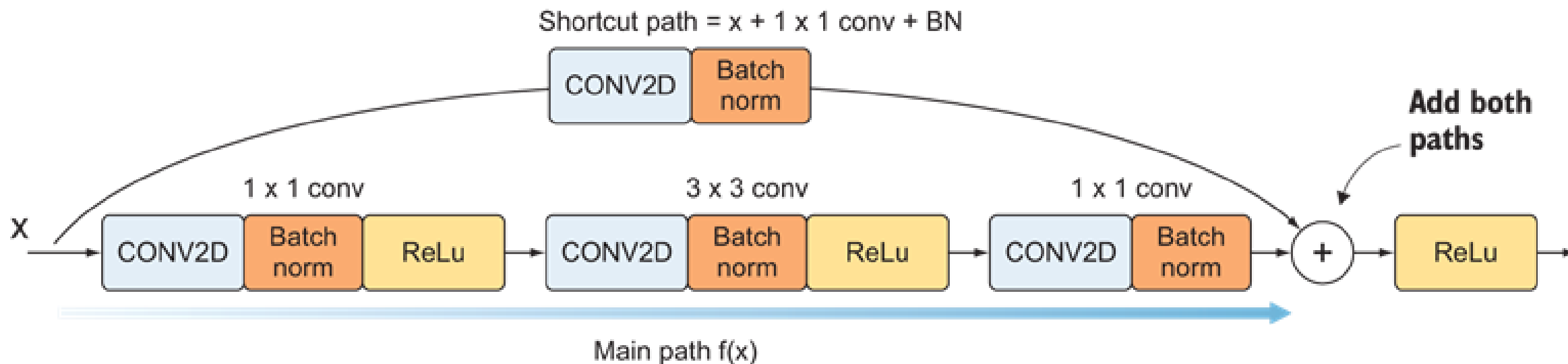
- 레이어를 건너 뛰는 경로를 만듦
- 매우 깊은 신경망의 경우에도 경사가 사라지지 않음 → 신경망의 깊이를 늘려도 학습이 잘 되므로 성능이 향상
- 최소한 이전 레이어에서 학습된만큼의 성능은 보장

ResNet에서 Skip Connection



ResNet에서 Skip Connection

Bottleneck residual block with reduce shortcut



전이 학습 Transfer Learning

- 근원 문제 `source`에 학습시킨 모델을 새로운 대상 문제 `target`에 적용하는 것
- 사전 학습: 근원 문제에 학습 시키는 것. 많은 데이터를 사용
- 미세 조정: 문제 대상 문제에 학습 시키는 것. 적은 데이터를 사용

전이학습이 해결하려는 문제

- 딥러닝은 일반적으로 많은 데이터가 필요
- 데이터를 수집하고 레이블링하는데 많은 비용이 소모
- 또한 딥러닝은 학습에 많은 계산 비용이 소모
- 만약 기존의 다른 문제에 학습시킨 모델을 약간의 미세 조정으로 높은 성능을 낼 수 있다면 데이터와 계산 비용을 절감할 수 있음

전이 학습의 방식

- 사전학습된 모델을 추가 학습 없이 그대로 사용
- 사전학습된 모델의 전반부를 특징 추출기로 사용 + 후반부에 분류기만 추가하고 추가된 부분만 학습
- 사전 학습된 부분을 포함해서 전체적으로 미세 조정

미세 조정

- 무작위로 초기화된 모형을 처음부터 학습시키는 것보다, 다른 문제에라도 학습시킨 모형을 추가 학습시키는 것이 학습에 유리할 가능성이 높음
- 이미지 처리 과제는 비슷한 특징들을 공유하므로 무작위로 초기화된 것보다는 유사할 가능성이 높음
- 일반적으로 근원 문제에 학습킬 때보다 작은 학습률을 사용

시나리오

- 대상 문제의 데이터가 적고, 근원 문제와 비슷한 경우
- 대상 문제의 데이터가 많고, 근원 문제와 비슷한 경우
- 대상 문제의 데이터가 적고, 근원 문제와 다른 경우
- 대상 문제의 데이터가 많고, 근원 문제와 다른 경우

시나리오#1: 적은 데이터, 비슷한 문제

- 데이터가 적기 때문에 특징 추출 부분까지 추가 학습을 하면 과적합의 위험이 있음
- 특징 추출 부분은 고정하고, 분류기만 추가 학습

시나리오#2: 많은 데이터, 비슷한 문제

- 데이터가 충분히 많다면 특징 추출 부분도 추가 학습을 시도해볼 수 있음
- 경우에 따라 특징 추출에 해당하는 레이어 모두가 아니라 그중 후반부의 일정 비율만 학습시키는 것도 가능

시나리오#3: 적은 데이터, 다른 문제

- 특징 추출 부분 중 특히 후반부는 추가 학습 필요
- 그러나 데이터가 충분치 않기 때문에 어려움

시나리오#3: 많은 데이터, 다른 문제

- 모형 전체를 추가 학습시키는 것도 고려해볼만
- 대부분의 경우 모형을 처음부터 학습시키는 것보다는 학습이 잘 됨
- 이미지 처리의 경우 아무리 다른 문제도 근본적으로는 비슷한 특징들을 공유