

# 신영

## 8.1 계층을 깊게 하는 이유

계층을 깊게 하는 방법

## 8.2 딥러닝의 역사

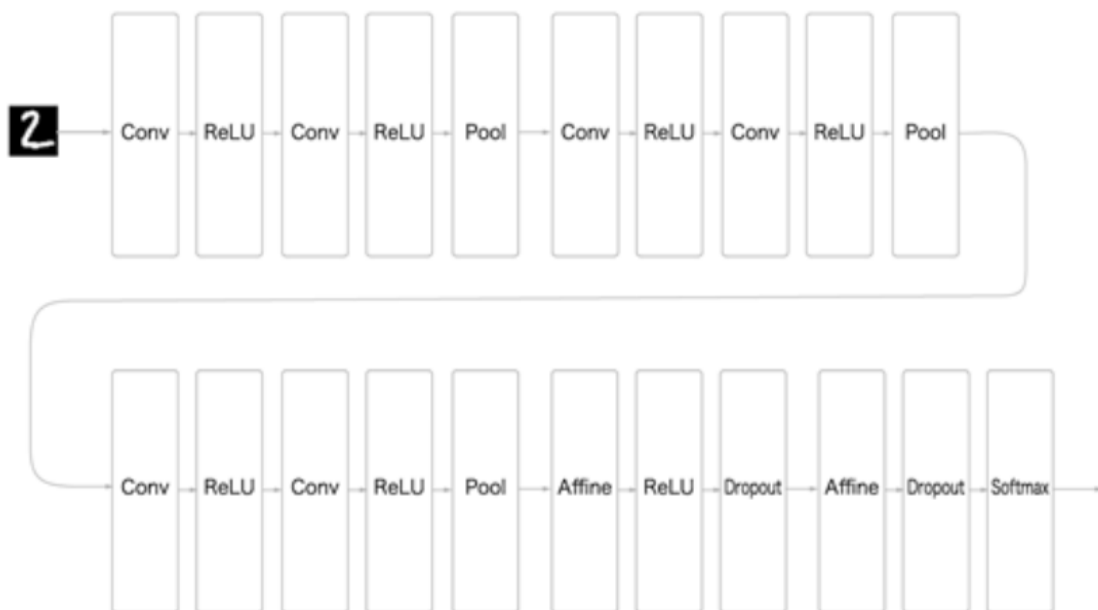
VGG

GoogleLeNet

ResNet

## 8.3 딥러닝의 속제와 활용

## 8.1 계층을 깊게 하는 이유



깊은 신경망을 써서 MNIST의 데이터셋에 사용해봤더니 학습 정확도가 99.38%로 매우 높게 되었다.

## 데이터 확장

입력 이미지를 알고리즘을 통해  
밝기, 확대, 축소, 미세하게 이동 등 인위적으로 수정을 가하고 이를 통  
해 이미지의 개수를 늘리는 기법이다.

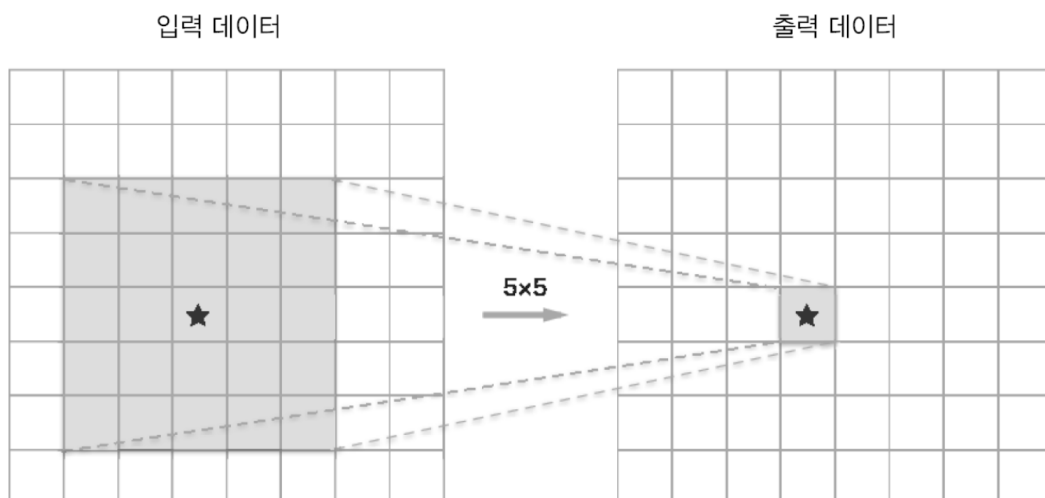
## 계층을 깊게 하는 방법

### 1. 작은 필터를 겹쳐주기

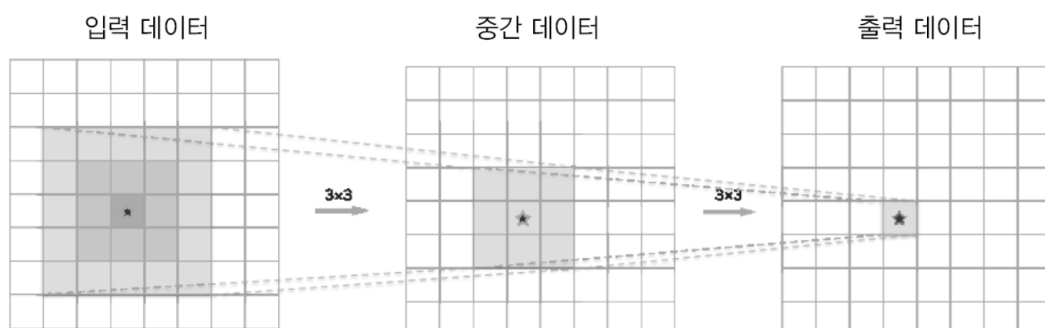
신경망 매개 변수의 개수가 감소한다.

이는 학습 효율성이 늘어난다.

- 1층을 사용한 경우



- 2층을 사용한 경우



(5×5)입력 데이터에  
(3×3), (3×3) 합성곱 연산을 2회 반복하면  
(5×5)를 한 번 연산한것과 같은 효과가 나온다.

이때, 연산량을 계산해보면

$5 \times 5 = 25$ 개,

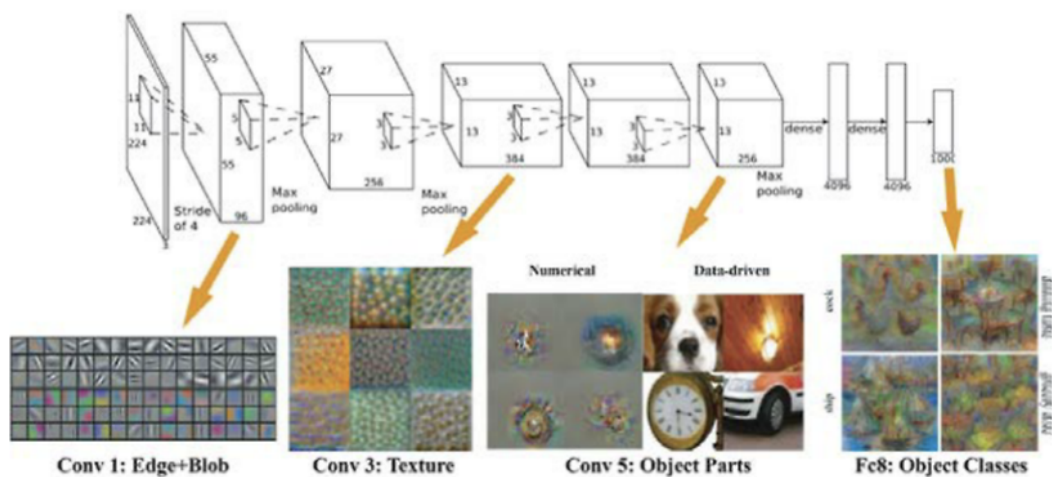
$2 \times 3 \times 3 = 18$ 개로 줄어든다.

즉, 작은 필터를 겹쳐서 여러 번 반복하게 되면

적은 매개 변수로 넓은 수용 영역(필터)을 커버할 수 있게 되었다.

## 특징을 뽑아내는 역할을 층마다 분배

신경망을 깊게 만들수록 문제를 단순화해서 학습 효율성이 증가한다.



만약 얇은 계층으로 이미지를 인식한다면 Conv계층에서 한번에 사물의 특징 대부분을 추출해야한다. 이걸 구현하려면, 학습 데이터가 많아야한다. 그러면 학습량도 많아지니 학습 시간이 오래걸린다.

반면, 계층을 깊게 하면 학습할 문제를 계층적으로 분해 가능하게 되므로, 즉 계층을 깊게 하면 각 층이 학습할 문제가 단순해지고, 문제를 풀기 쉬운 단순한 문제로 분해 가능하니 학습 효율이 좋아진다.

## 활성화 함수를 통해 '비선형' 적용

### ReLU함수를 통해 비선형 부여

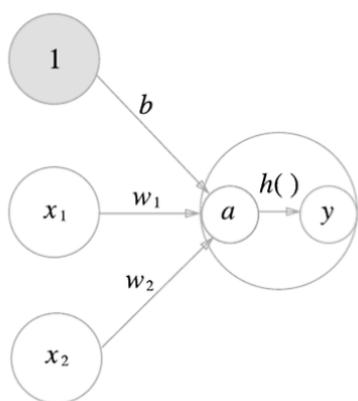


그림 3-7 시그모이드 함수의 그래프\*

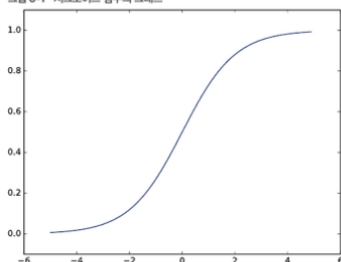
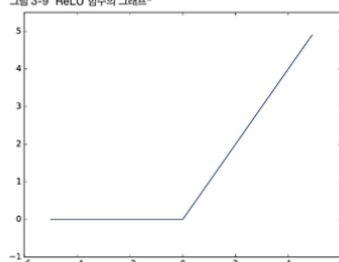


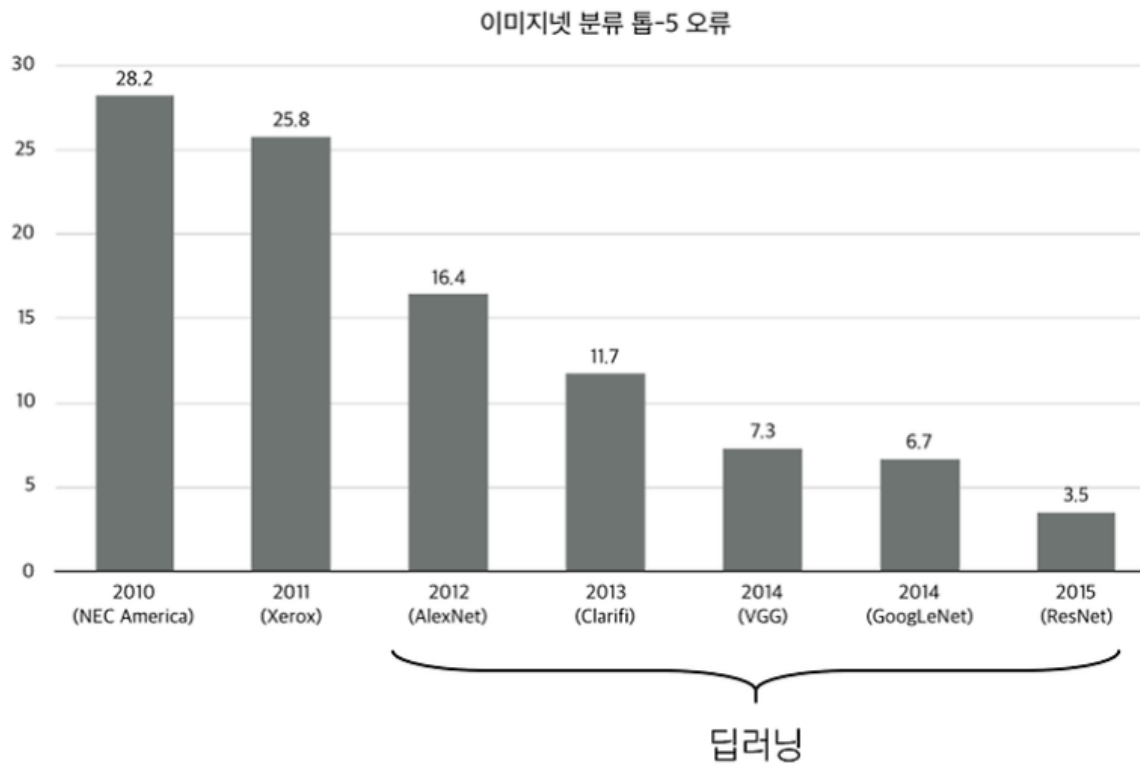
그림 3-9 ReLU 함수의 그래프\*



ReLU등의 활성화 함수를 합성곱 계층 사이에 끼움으로써 신경망의 표현력을 개선시킬 수 있다.

이는 **ReLU 함수를 통해 비선형** 힘을 가하고, 이러한 비선형 함수가 겹치면서 더 복잡한 것도 표현할 수 있게 된다.

## 8.2 딥러닝의 역사



ILSVRC라는 이미지 인식 기술을 겨루는 대회에서 2012년 AlexNET이 등장한 이후, 2012년~2015년까지 AlexNET, VCG, GoogleNet, ResNet이 모두 딥러닝을 써서 우승했다.

이때 ResNet은 이미지 인식 오차가 3.5%로 인간의 인식 능력과 비슷하다.

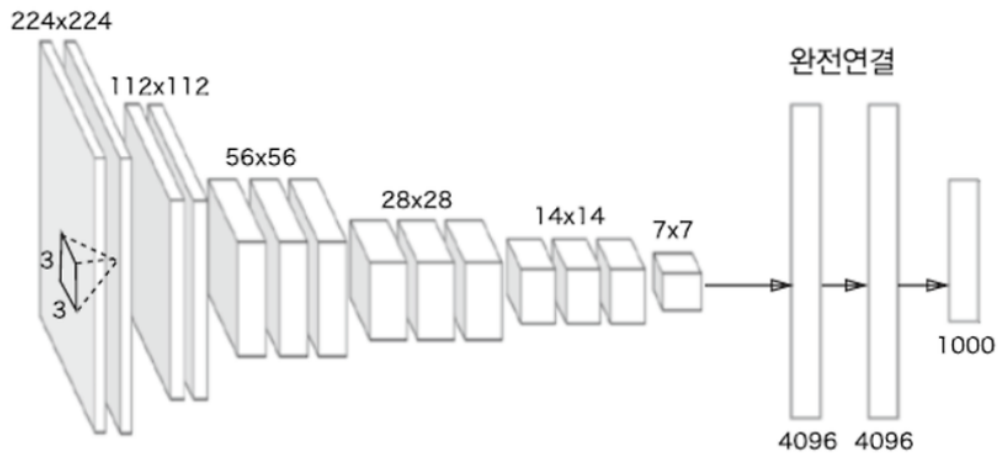
## VCG

**합성곱 계층(Convolutional Layer) : 3x3 필터**

**풀링 계층(Pooling Layer)**

**완전연결계층 (Affine Layer = Fully Connected Layer)**

GoogleNet과 성능이 크게 뒤쳐지지 않고, 구조가 심플해서 많이 사용



## GoogleLeNet

**합성곱 계층** : 다양한 크기의 필터 적용

### 풀링 계층

- 크기를 줄여줘서 밀도 높은 연산이 가능해지고, 학습할 parameter도 감소하며, 모델의 이동 불변성(어디에 물체가 있는지 인식하는 능력) 특징을 가질 수 있다.

### 인셉션 구조 : 필터 결합

- (1×1), (3×3), (5×5), (3×3 max pooling)을 거쳐 여러 특징을 추출하고 이를 합친 최종 필터맵을 다음 계층으로 전달
- 이때, (1×1) 합성곱 연산은 채널 차원의 수를 줄여서 연산량을 줄여 줌.

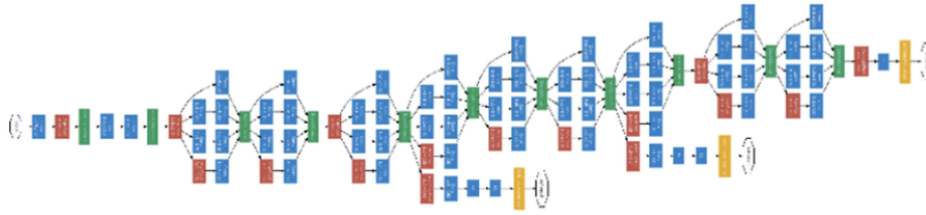
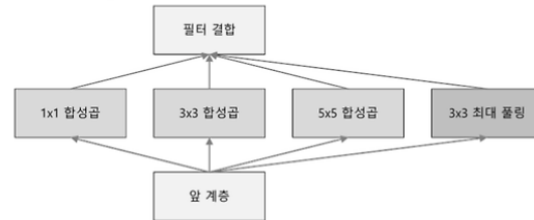


그림 8-11 GoogLeNet의 인셉션 구조<sup>[23]</sup>



중간에 노란색 노드는 일반적으로 출력층을 나타내며

깊은 신경망 중간중간에 있는 이유는 보조 분류기의 역할을 하기 때문이다.

학습이 완료된 후에는 이 보조 분류기를 제거하여 최종 예측에 영향을 끼치지 않게 해준다.

## ResNet

합성곱 계층 (Weight Layer) : 주로 3x3 필터로 합성곱 연산

ReLU계층 : 기울기 소실 문제 완화

스킵 연결(Skip Connection)

- 잔차 연결 : 기울기 소실(Vanishing Gradient) 문제 해결
  - 잔차 연결(Residual Learning)을 통해 기울기가 이전 층으로 직접 전달되도록 (ResNet은 합성곱 2개 계층마다) 하여 기울기 소실 문제를 해결

그림 8-12 ResNet의 구성요소<sup>[24]</sup> : 'weight layer'는 합성곱 계층을 말한다.

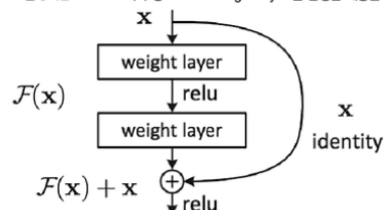
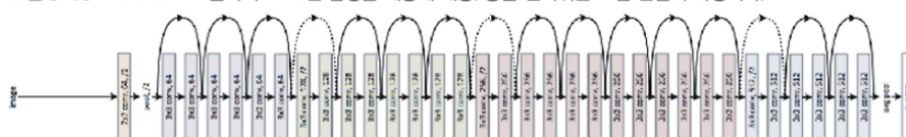


그림 8-13 ResNet<sup>[24]</sup> : 블록이 3x3인 합성곱 계층에 대응. 층을 건너뛰는 스킵 연결이 특징이다.



## 8.3 딥러닝의 속제와 활용

### 딥러닝의 속제

딥러닝은 Conv 계층에서 대부분의 시간(약 89% ~ 95%)를 소비한다.

**즉 Conv 계층의 연산을 효율적으로 해야 한다.**

(이를 해결하기 위해 GPU, 분산 학습 등의 방법이 등장했다)

연산 능력 외에, **메모리 용량과 버스 대역폭 능력이 필요하다.** 메모리 용량면에서는

대량의 가중치 매개 변수와 중간 데이터를 저장할 수 있어야 하고, 버스 대역폭면에서는 GPU의 버스에 데이터를 흐를 수 있게 해야한다.

### 딥러닝 활용

#### 사물 검출

이미지 속의 사물의 위치와 종류를 알아내는 기술

사물 인식은 이미지 전체가 대상이지만, 사물 검출은 이미지 어딘가에 있는 사물의 위치를 파악해야한다.

CNN을 이용하여 사물 검출하는 방식 중, R-CNN이 유명하다.

#### 분할(Segementaion)

분할이란, 이미지를 픽셀 수준에서 분류하는 문제이다.

앞에서 CNN을 사용할 때는, 이미지 전체에서 추론하는 문제를 했는데 픽셀 단위로 추론하면 계산량이 엄청 많다.



분할을 수행하려면 FCN(Fully Convolutional Network)를 사용한다.

### **사진 캡션 생성**

컴퓨터 비전과 자연어를 융합한 연구로써, 사진을 신경망에 입력 시, 그 사진을 설명하는 글(사진캡션)을 자동으로 생성하는 연구이다.

딥러닝으로 사진 캡션을 추가하려면, NIC 모델을 사용한다. NIC 모델은, CNN으로 사진의 특징을 추출 후, RNN에 넘겨서, 텍스트를 순환적으로 생성하는 방법이다.

### **자율 주행**

자율 주행은 경로 탐색 기술 + 카메라 + 레이저 등의 기술을 쓰지만, 무엇보다 주위를 올바르게 인식하는 기술이 중요하다.

주위 환경을 인식하는 SegNet이라는 CNN기반 신경망을 사용한다.

SegNet은 Segementation처럼, 입력 이미지를 분할(픽셀 수준에서 판정)함으로써, 여러 사물을 판별한다.