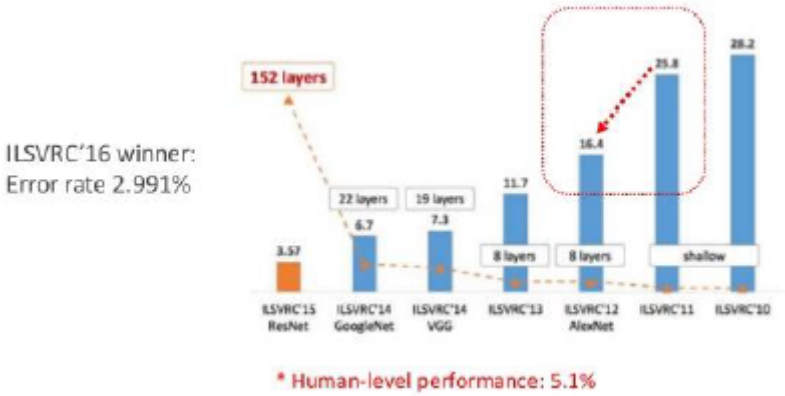
딥러닝

컨볼루션 신경망 사례연구

- **영상분류** : 도전적 문제
  - **ImageNet** 데이터베이스
    - 2만 2천여 부류에 대해 부류별로 수백~수만장의 영상을 인터넷에서 수집하여 1500만여 장의 영상을 구축하고 공개
  - ILSVRC 대회
    - 1000부류에 대해 분류, 검출, 위치 지정 문제 : 1순위와 5순위 오류율로 대결
    - 120만 장의 훈련집합, 5만 장의 검증집합, 15만 장의 테스트집합
    - 우승 : AlexNet -> Clarifi -> GoogLeNet & VGGNet -> ResNet
  - 우승한 CNN은 프로그램과 가중치를 공개함으로써 널리 사용되는 표준 신경망이 됨
- 예시

- 성능 발전



AlexNet

구조

- **컨볼루션층 5개, 완전연결층 3개**
  - 8개 층에 290400 - 186624 - 64896 - 43264 - 4096 - 4096 - 1000개의 노드 배치
- 컨볼루션층은 200만개, FC층은 6500만개 가량의 매개변수
  - FC층에 30배 많은 매개변수 : **향후 CNN은 매개변수를 줄이는 방향으로 발전**

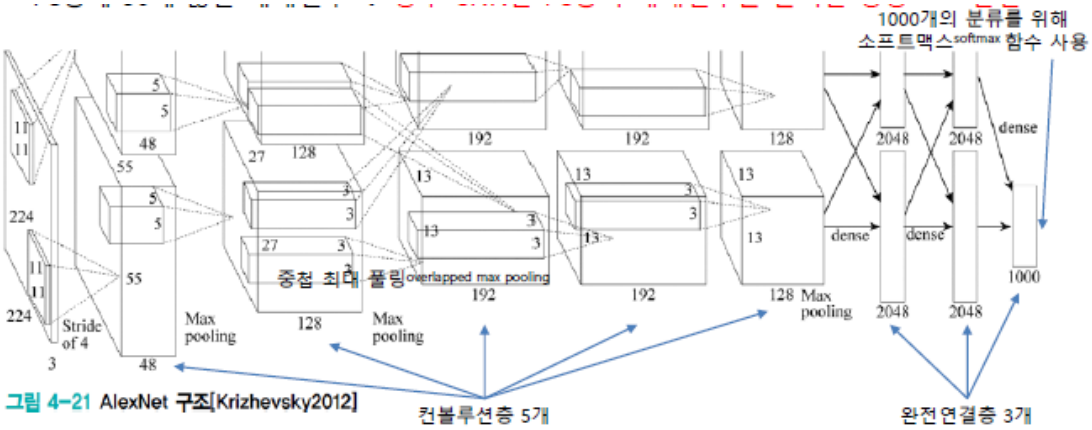


그림 4-21 AlexNet 구조[Krizhevsky2012]

- GPU 메모리 크기 제한으로 인해 GPU#1, GPU#2로 분할하여 학습 수행
- 첫번째 컨볼루션 층 : 큰 보폭으로 다운샘플링(풀링 X)
- 3번째 컨볼루션 층 : GPU#1과 GPU#2의 결과를 함께 사용
- 마지막 층 : 소프트맥스 함수 사용 1000개의 분류

성공 요인

- 외적 요인
  - **ImageNet**이라는 대용량 데이터베이스
  - **GPU**를 사용한 병렬처리
- 내적 요인
  - 활성함수로 **ReLU** 사용
  - **지역 반응 정규화** 기법 적용
    - 1번째, 3번째 최대 풀링 전 적용
  - 과잉적합 방지하는 여러 규제 기법 적용
    - **데이터 확대**(잘라내기, 반전으로 2048배 확대)
    - **드롭아웃**등 (완전연결층에서 사용)
- 테스트 단계에서 앙상블 적용
  - 입력된 영상을 잘라내기와 반전을 통해 증가하고, 증가된 영상들의 예측 평균으로 최종 인식
  - 2~3%만큼 오류율 감소 효과

VGGNet

핵심 아이디어

- 3*3의 **작은 커널** 사용
- **신경망을 더욱 깊게 만듬**(깊이가 어떤 영향을 주는지 확인)
- 컨볼루션층 8~16개를 두어 AlexNet의 5개에 비해 2~3배 깊어짐
- 16층짜리 VGG-16(CONV 13 + FC 3)

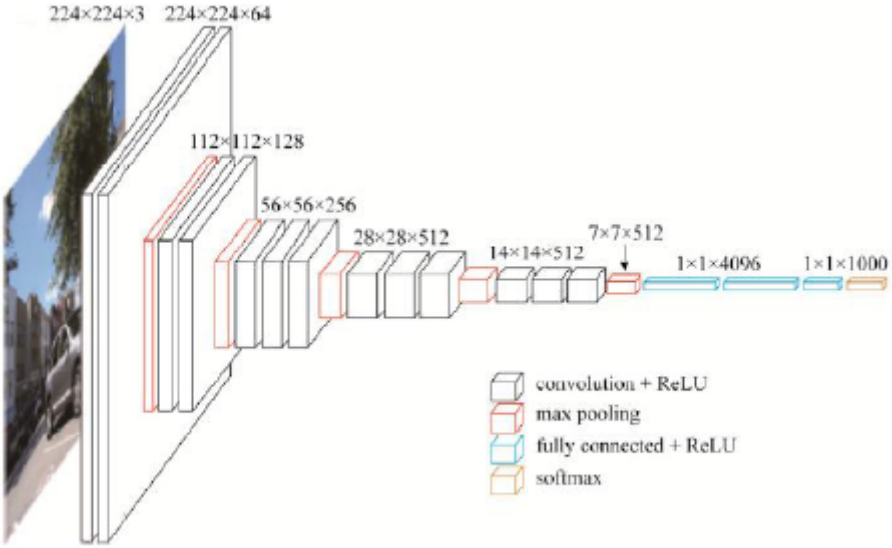
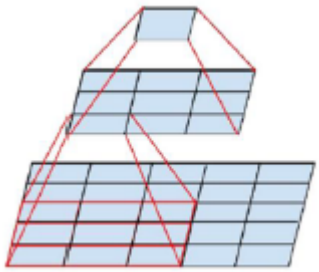


그림 4-22 VGGNet 구조[Simonyan2015]

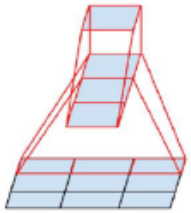
작은 커널

- **작은 커널**의 이점 ▶ GoogLe의 인셉션 모듈에 영향
  - 큰 크기의 커널은 여러 개의 작은 크기 커널로 분해가능
    - 매개변수의 수는 줄어들면서 신경망은 깊어짐
  - 5*5 커널을 2층의 3*3 커널로 분해하여 구현



매개변수
5*5 커널인 경우, 25
3*3 커널인 경우, 9+9 =18

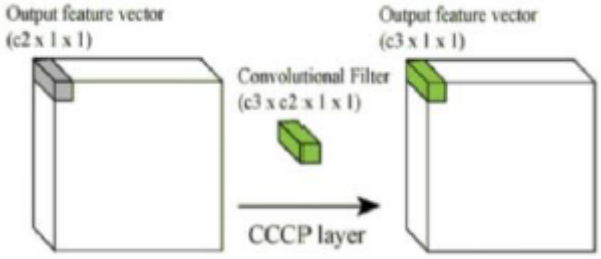
  - 3*3 커널을 1*3 커널과 3*1 커널로 분해하여 구현



매개변수
3*3 커널인 경우, 9
1*3 커널, 3*1인 경우, 3+3 =6

→ 유사하게 n*n 커널은 1*n 커널과 n*1 커널로 분해될 수 있으며, n이 클수록 매개변수의 수는 줄어드는 효과가 큼

- **1*1 커널**
  - **차원 축소 효과**
    - c2 > c3 : **차원 축소**(연산량 감소), 깊은 신경망



  - m*n의 특징 맵 8개에 1*1 커널을 4개 적용 -> m*n의 특징 맵 4개가 됨
    - 8*m*n 텐서에 8*1*1 커널을 4개 적용하여 4*m*n 텐서를 출력하는 셈

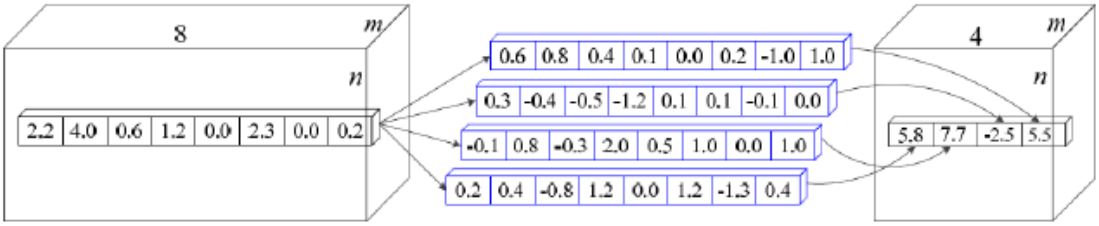


그림 4-23 1*1 컨볼루션 예제

- ReLU와 같은 비선형 활성함수를 적용하면 특징 맵의 분별력 증가
- 네트워크 속의 네트워크(NIN)에서 유래
- VGGNet은 적용 실험을 하였지만, 최종 선택하지는 않음(GoogLeNet에서 사용)

GoogLeNet

NIN 구조

- **Mlpconv층**이 컨볼루션 연산을 대신함 -> **비선형 특성**을 잘 **표현**하기 위함
- Mlpconv는 커널을 옮겨가면서 MLP의 전방 계산을 수행
- MLP 또한 컨볼루션 연산처럼 오류 역전파를 통해 학습 가능

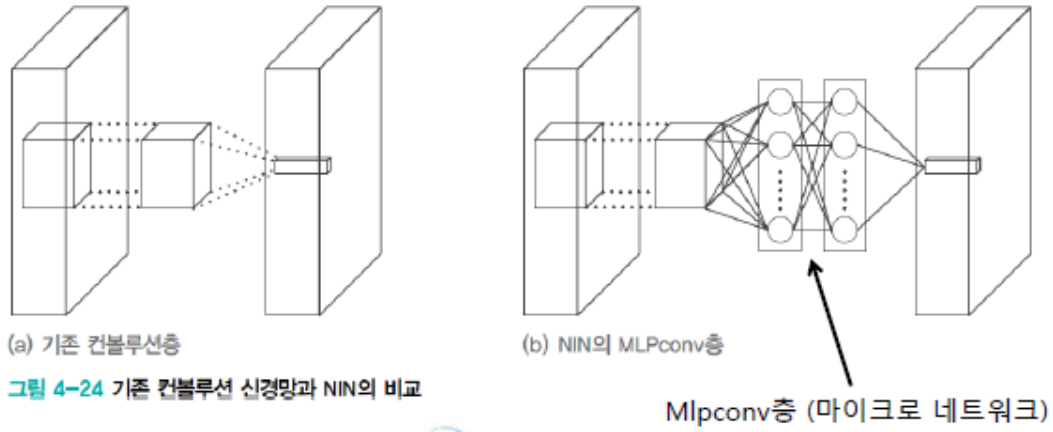


그림 4-24 기존 컨볼루션 신경망과 NIN의 비교

- NIN이 사용하는 전역 평균 풀링
  - VGGNet의 완전 연결층
    - 1억 2천 2백만 개의 매개변수(과잉적합의 원인)
  - **전역 평균 풀링**(GAP)
    - Mlpconv가 부류 수만큼 특징 맵을 생성하면, 특징 맵 각각을 평균하여 출력 노드에 입력 -> 이 방식으로 **매개변수를 줄임**

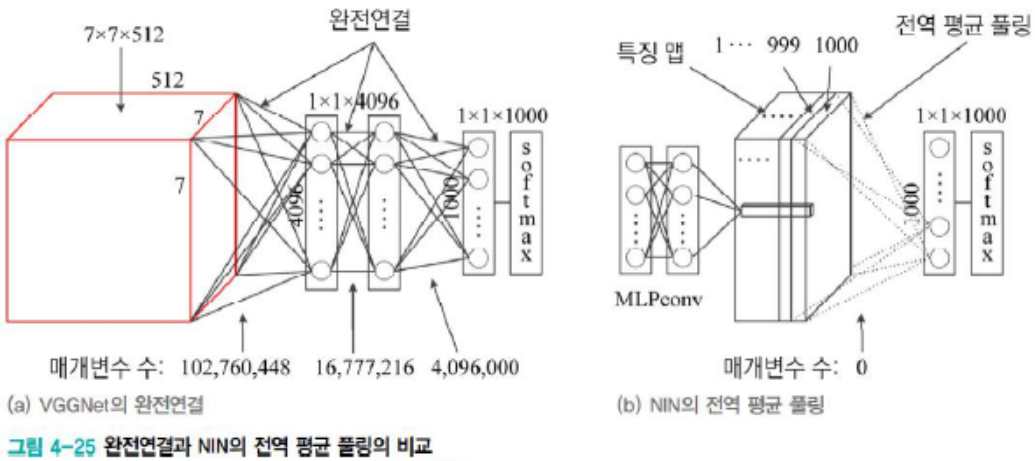


그림 4-25 완전연결과 NIN의 전역 평균 풀링의 비교

인셉션 모듈

- 수용장에서 **더 다양한 특징을 추출**하기 위해 NIN의 구조를 확장하여 **여러 개의 병렬적인 컨볼루션 층**을 가지도록 함
- 마이크로 네트워크로 MIpconv 대신 **네 종류의 컨볼루션 연산** 사용 -> **다양한 특징 추출**
- 1*1 컨볼루션을 사용해 차원 축소
    - 매개변수의 수를 줄임 + 깊은 신경망
- 3*3, 5*5 같은 다양한 크기의 컨볼루션들을 통해서 다양한 특징들을 추출

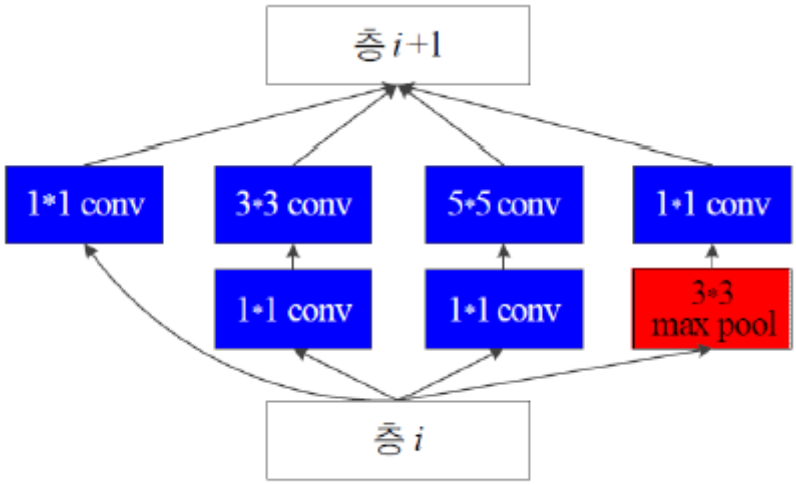


그림 4-26 GoogLeNet의 인셉션 모듈

- 인셉션 모듈(I)를 9개 결합한 GoogLeNet

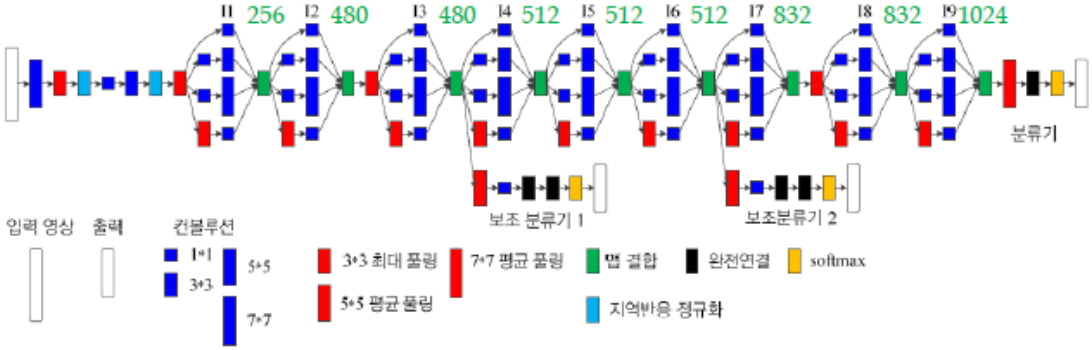


그림 4-27 GoogLeNet의 구조

    - 매개변수가 있는 층은 22개, 없는 층은(풀링) 5개로 총 27개 층
    - 완전 연결층은 1개에 불과
        - 1백만 개의 매개변수, VGGNET에 비하면 1%에 불과
- **보조 분류기**
    - 원 분류기의 오류 역전파 결과와 보조 분류기의 오류 역전파 결과를 결합함으로써 **그레이디언트 소멸 문제 완화**
    - 학습시에만 도우미 역할을 하고, 동작시에는 제거됨

ResNet

- **잔류학습**이라는 개념을 이용하여 성능 저하를 피하면서 **층 수를 대폭 늘림**(최대 1202층)
    - 깊은 신경망일수록 데이터의 대표적인 특징을 잘 추출할 수 있음
- 원래 컨볼루션 신경망

$$\mathbf{F}(\mathbf{x}) = \tau(\mathbf{x} \circledast \mathbf{w}_1) \circledast \mathbf{w}_2$$
$$\mathbf{y} = \tau(\mathbf{F}(\mathbf{x}))$$

- 잔류 학습은 지름길 연결된 x를 더한 F(x) + x에 τ를 적용, F(x)는 잔류

$$y = \tau(F(x) + x)$$

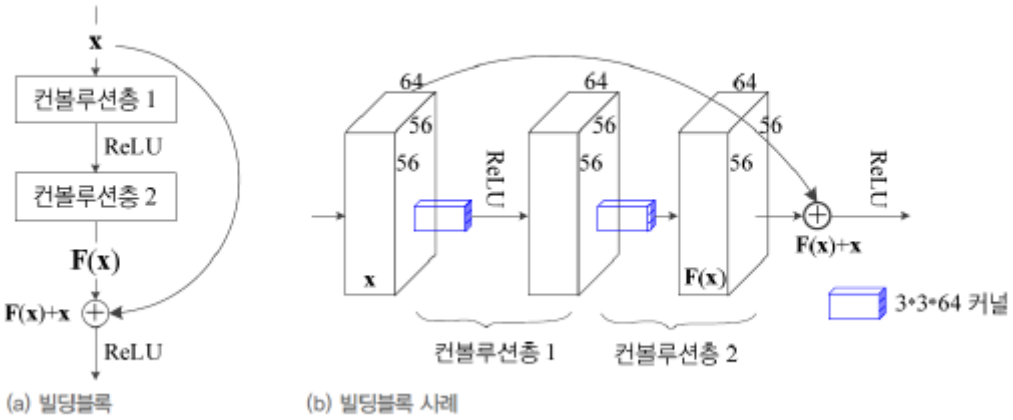


그림 4-28 잔류 학습의 구조와 동작

- 지름길 연결을 두는 이유
  - 그레디언트 소멸 문제 해결

식 (4.14)의 그레이디언트 식에서 $\frac{\partial}{\partial \mathbf{x}_l}\sum_{i=l}^{L-1}\mathbf{F}(\mathbf{x}_i)$ 이 -1이 될 가능성이 거의 없음

$$\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\frac{\partial \mathbf{x}_L}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\left(1 + \frac{\partial}{\partial \mathbf{x}_l}\sum_{i=l}^{L-1}\mathbf{F}(\mathbf{x}_i)\right) \qquad (4.14)$$

- 34층짜리 ResNet 예시

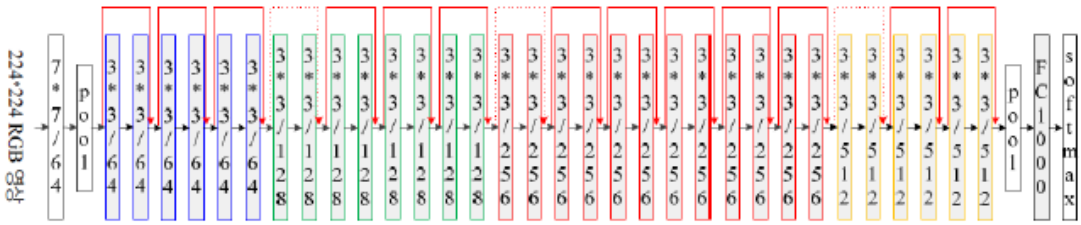


그림 4-29 ResNet 예제(34층)

- VGGNet과 같은 점 : 3*3 커널 사용
- VGGNet과 다른 점
  - 잔류 학습 사용
  - 전역 평균 풀링 사용(FC층 제거)
  - 배치 정규화 적용(드롭아웃 적용 불필요)

ILSVRC 대회 성적

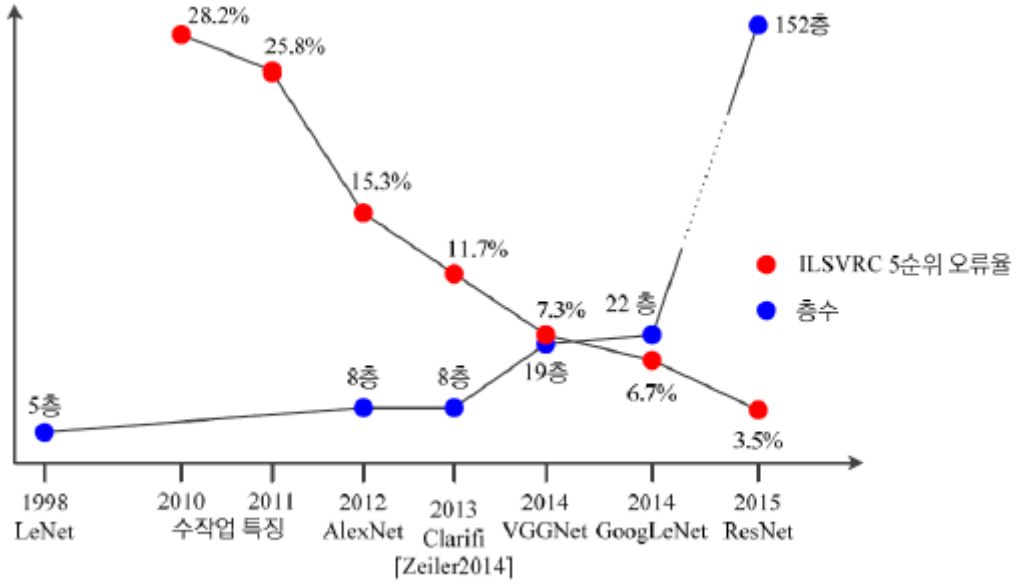


그림 4-30 CNN의 발전 추세

- 분류 문제는 성능 포화(사람 성능에 필적)
- 물체 검출 문제에 집중

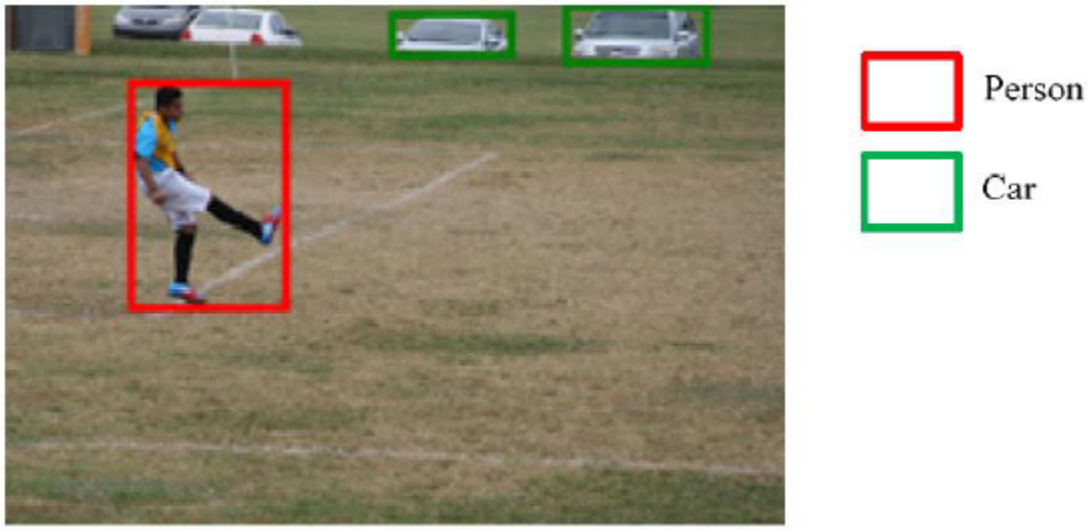


그림 4-31 ILSVRC 물체 검출 문제

생성 모델

- 사람의 생성 모델 : 세상에 나타나는 현상을 오랫동안 지켜보면서 학습한 결과
  - 기계학습이 훈련집합을 사용해 비슷한 생성 모델을 구축할 수 있다면 강한 인공지능에 한발
- 생성 모델은 분별 모델에 비해 데이터 생성 과정에 대한 보다 깊은 이해를 필요로 함

생성 모델이란?

표 4-1 분별 모델과 생성 모델

모델	학습 단계가 할 일	예측 단계가 할 일	지도 여부
분별 모델	$P(y\|\mathbf{x})$ 추정	$f:\mathbf{x}\mapsto y$	지도 학습
생성 모델	$P(\mathbf{x})$ 또는 $P(\mathbf{x}\|y)$, $P(\mathbf{x},y)$ 추정	$f:$ 씨앗 $\mapsto \mathbf{x}$ 또는 $f:$ 씨앗 $y\mapsto \mathbf{x}$, $f:$ 씨앗 $\mapsto \mathbf{x},y$	비지도 학습

예제 4-1 **생성 모델과 분별 모델의 확률분포 추정과 예측**

특징 벡터가 2차원이고 이진값을 가지며, 부류가 2개라 가정하자. 훈련집합은 다음과 같다.

$$\mathbb{X} = \left\{\binom{0}{0}, \binom{0}{0}, \binom{0}{1}, \binom{0}{1}, \binom{0}{1}, \binom{1}{0}, \binom{1}{1}, \binom{1}{1}, \binom{1}{1}, \binom{1}{1}\right\}, \mathbb{Y} = \{1,1,1,0,1,0,0,0,0,1\}$$

생성 모델이 추정하는 확률 분포

$P(\mathbf{x}) =$

$\mathbf{x} = (0,0)^T$	0.2
$\mathbf{x} = (0,1)^T$	0.3
$\mathbf{x} = (1,0)^T$	0.1
$\mathbf{x} = (1,1)^T$	0.4

$P(\mathbf{x}|y) =$

	$y = 0$	$y = 1$
$\mathbf{x} = (0,0)^T$	0.0	0.4
$\mathbf{x} = (0,1)^T$	0.2	0.4
$\mathbf{x} = (1,0)^T$	0.2	0.0
$\mathbf{x} = (1,1)^T$	0.6	0.2

$P(\mathbf{x}, y) =$

	$y = 0$	$y = 1$
$\mathbf{x} = (0,0)^T$	0.0	0.2
$\mathbf{x} = (0,1)^T$	0.1	0.2
$\mathbf{x} = (1,0)^T$	0.1	0.0
$\mathbf{x} = (1,1)^T$	0.3	0.1

분별 모델이 추정하는 확률 분포

$P(y|\mathbf{x}) =$

	$y = 0$	$y = 1$
$\mathbf{x} = (0,0)^T$	0.0	1.0
$\mathbf{x} = (0,1)^T$	0.33	0.67
$\mathbf{x} = (1,0)^T$	1.0	0.0
$\mathbf{x} = (1,1)^T$	0.75	0.25

학습을 마쳤으니, 이제 예측 단계를 수행해보자. 생성 모델이 $P(\mathbf{x})$를 사용하고, 네 가지 $\mathbf{x}$ 값의 확률에 따라 $\binom{0}{0}$에게 [0.0,0.2], $\binom{0}{1}$에게 (0.2,0.5], $\binom{1}{0}$에게 (0.5,0.6], $\binom{1}{1}$에게 (0.6,1.0] 구간을 부여하자. 난수로 0.34가 나오면 (0.2,0.5]에 속하므로 $\binom{0}{1}$을 생성하고, 0.83이 나오면 (0.6,1.0]에 속하므로 $\binom{1}{1}$을 생성한다.

분별 모델의 예측을 생각해보자. 만일 테스트 샘플 $\binom{0}{1}$이 주어지면, $P\left(y = 0|\mathbf{x} = \binom{0}{1}\right) = 0.33$ 이고 $P\left(y = 1|\mathbf{x} = \binom{0}{1}\right) = 0.67$이므로 $y = 1$이라고 분류하면 된다.

- 실제 상황에서 생성 모델
  - 현실에 내재한 데이터 발생 분포 $P_{data}(x)$ -> 알아낼 수 없음
  - $P_{data}(x)$를 모방하는 모델의 확률 분포 $P_{model}(x;\theta)$
    - $P_{model}(x;\theta)$를 명시적으로 추정하는 것도 불가능
    - 현대 기계 학습은 주로 딥러닝 모델을 사용하여 확률 분포를 암시적으로 표현
      - GAN, VAE, RBM

GAN(Generative Adversarial Network)

- 우월한 성능
  - 사람을 상대로 진짜와 가짜 구별하는 실험에서
    - MNIST 52.4%, CIFAR-10 78.7%(50%이면 완벽히 속임)
- 아이디어
  - 생성기 G와 분별기 D의 대립 구도
    - G는 가짜 샘플 생성(위조지폐범)

        - D는 가짜와 진짜를 구별(경찰)
    - GAN의 목표는 **위조지폐범의 승리**
        - **G**가 만들어내는 샘플을 D가 구별하지 못하는 수준까지 학습

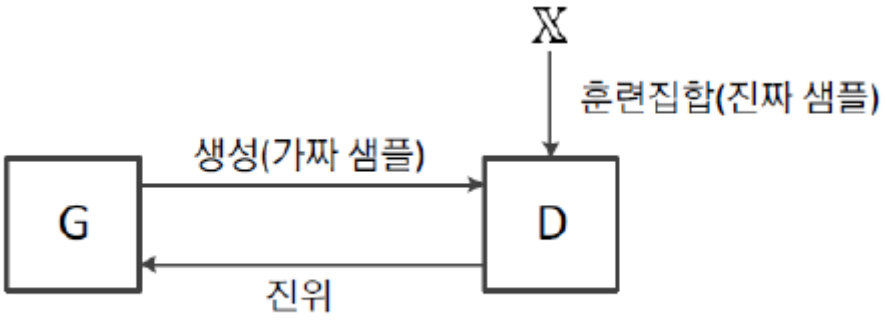


그림 4-34 GAN의 원리

딥러닝은 왜 강력한가?

- 종단간(end-to-end) 최적화된 학습 기능
    - 고전적인 방법
        - 분할, 특징 추출, 분류를 따로 구현한 다음 이어 붙임
        - 사람의 직관에 따르므로 성능에 한계
        - 인식 대상이 달라지면 새로 설계 필요

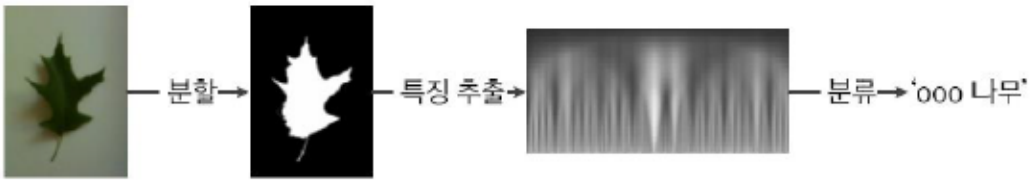


그림 4-37 여러 단계를 따로 설계 구현하는 고전적인 접근방식(나뭇잎 인식 사례)

    - 딥러닝은 전체 과정을 동시에 최적화(통째 학습이라 부름)

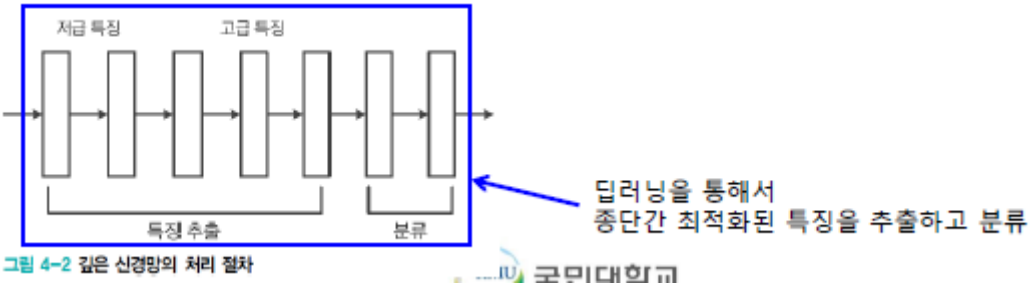


그림 4-2 깊은 신경망의 처리 절차

- 깊이의 중요성

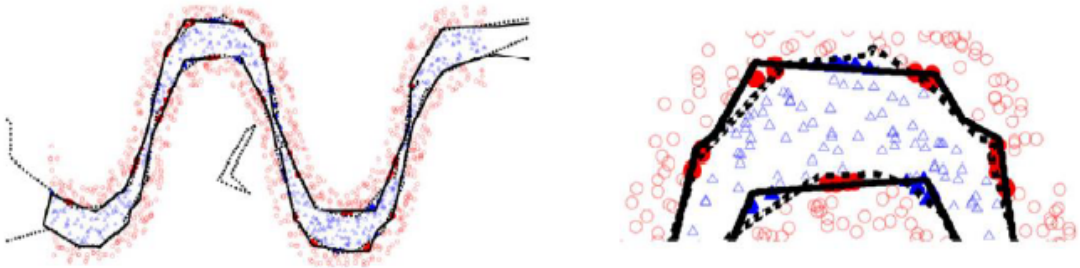


그림 4-38 은닉층의 개수가 늘어남에 따른 표현력 증가

    - 점선은 20개 노드를 가진 **은닉층 하나** 짜리 신경망
    - 실선은 각각 10개 노드를 가진 **은닉층 2개** 짜리 신경망(**더 정교한 분할**)
- 계층적 특징

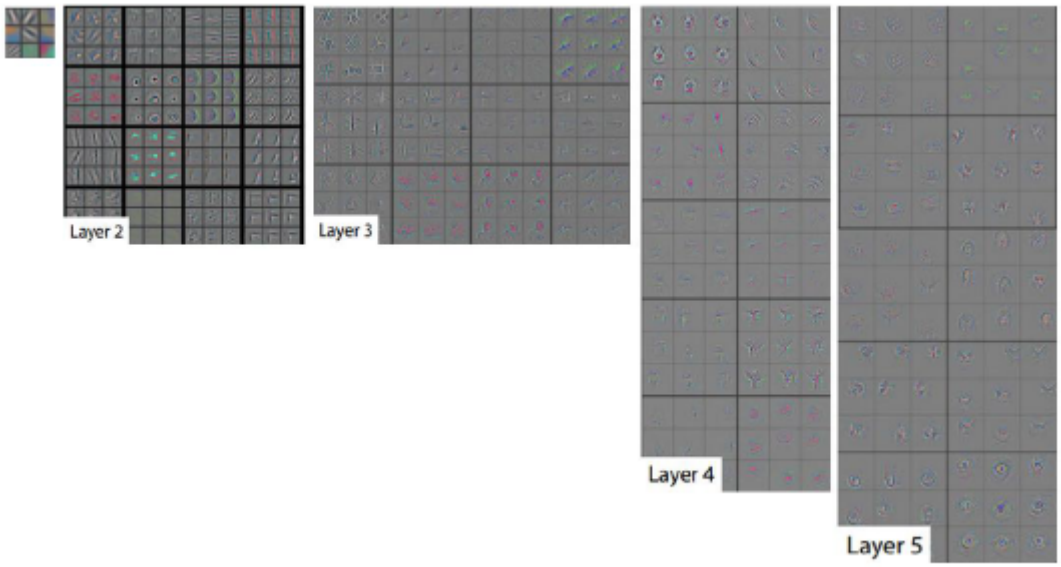


그림 4-40 CNN의 계층적 특징 추출

- [그림 4-40]은 ImageNet으로 학습한 특징맵
  ▶ 계층 구조, 깊은 신경망에서는 층의 역할이 잘 구분됨
- 얕은 신경망은 하나 또는 두 개의 은닉층이 여러 형태의 특징을 모두 담당