

인 공 지 능

[심층학습 기초 III]

본 자료는 해당 수업의 교육 목적으로만 활용될 수 있음.
일부 내용은 다른 교재와 논문으로부터 인용되었으며, 모든 저작권은 원 교재와 논문에 있음.

4.4 컨볼루션 신경망 사례연구

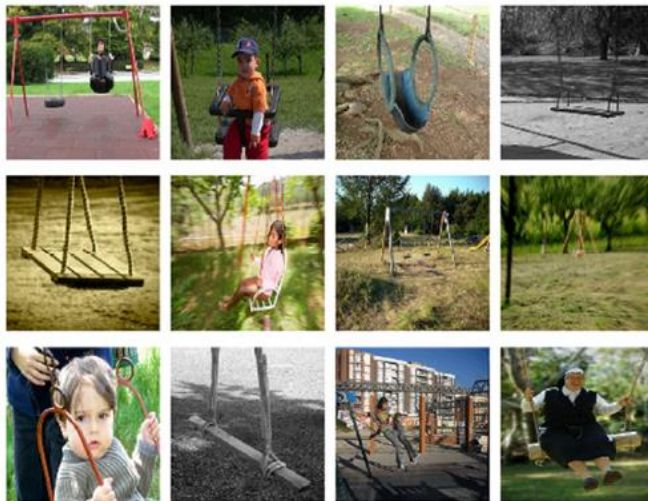
- 4.4.1 AlexNet
- 4.4.2 VGGNet
- 4.4.3 GoogLeNet
- 4.4.4 ResNet

4.4 컨볼루션 신경망 사례연구

■ 영상 분류 image classification: 과거에는 매우 어렵고 도전적인 문제였음

■ ImageNet

- 2만 2천여 부류에 대해 부류별로 수백~수만장의 사진을 인터넷에서 수집하여 1500만여 장의 사진을 구축하고 공개 (현재, 부류와 개수가 추가됨)
- ILSVRC ImageNet Large Scale Visual Recognition Competition 대회 (CVPR 학술대회에서 개최)
 - 1000가지 부류에 대해 분류, 검출, 위치 지정 문제: 1순위와 5순위 오류 성능 대결
 - 120만 장의 훈련집합, 5만 장의 검증집합, 15만 장의 테스트집합
 - 우승: AlexNet (2012) → Clarifai팀 (2013) → GoogLeNet&VGGNet (2014) → ResNet (2015)
- 우승한 모델은 코드와 학습된 가중치를 공개함으로써 널리 사용되는 표준 신경망이 됨



(a) 'swing' 부류



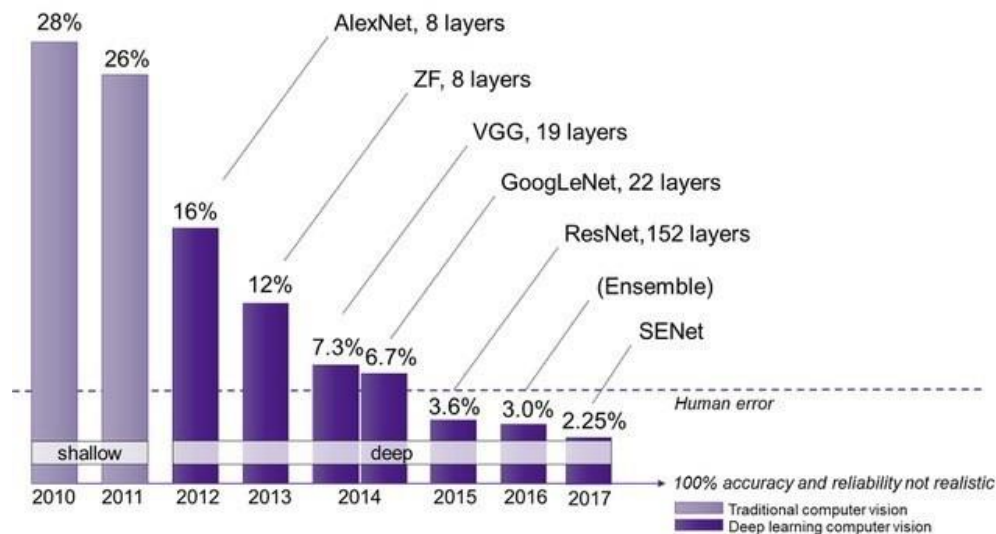
(b) 'Great white shark' 부류

4.4 컨볼루션 신경망 사례연구

■ ILSVRC 영상 분류 예측 결과 예시



■ ILSVRC 결과



4.4.1 AlexNet

■ 구조

- 컨볼루션층 5개와 완전연결 fully connected (FC) 층 3개
 - 8개 층에 290400-186624-64896-43264-4096-4096-1000개의 노드 배치
- 컨볼루션층은 200만개, FC층은 6500만개 가량의 매개변수
 - FC층에 30배 많은 매개변수 → 향후 CNN은 FC층의 매개변수를 줄이는 방향으로 발전

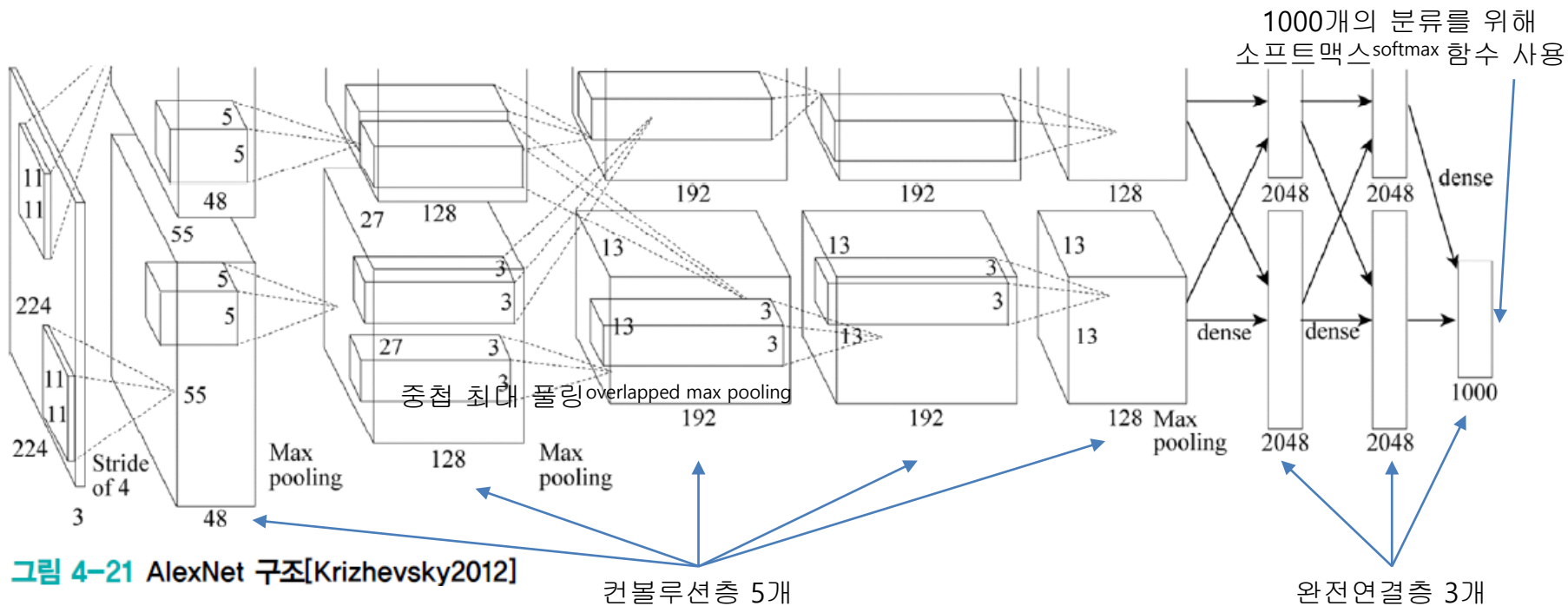
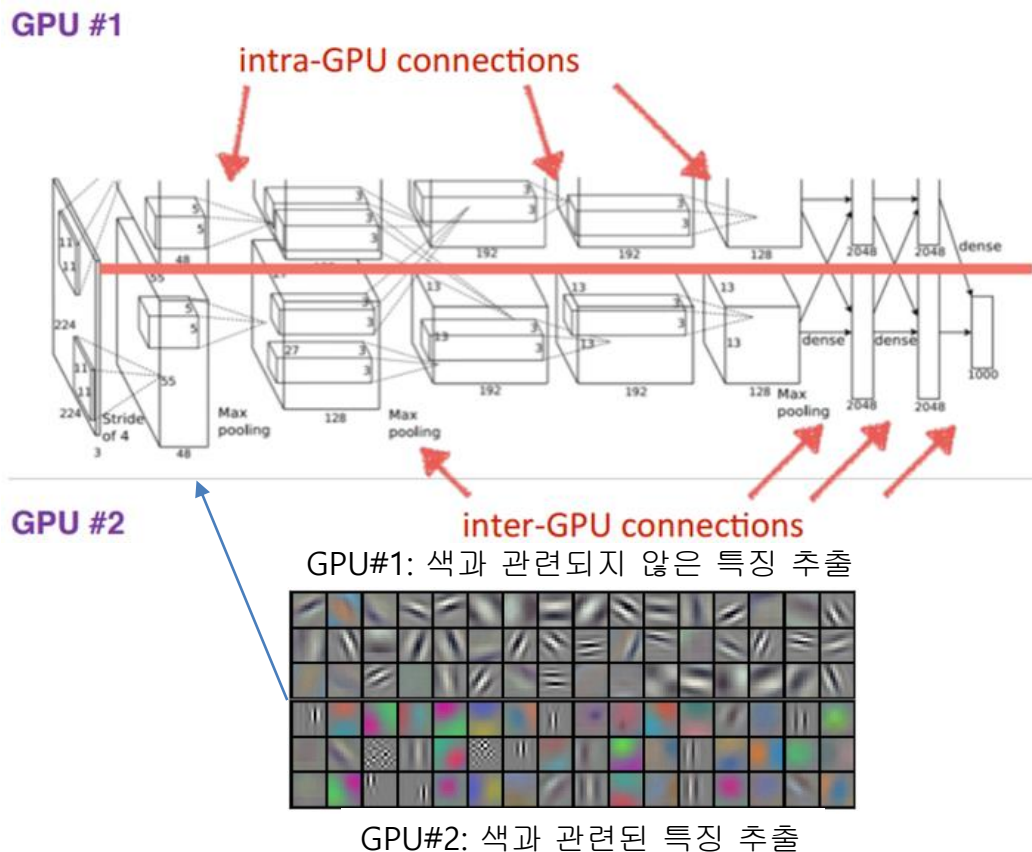


그림 4-21 AlexNet 구조[Krizhevsky2012]

4.4.1 AlexNet

■ 구조

- 당시 GPU의 메모리 크기 제한으로 인해 GPU#1, GPU#2으로 분할하여 학습 수행
 - 3번째 컨볼루션 층은 GPU#1과 GPU#2의 결과를 함께 사용 (inter-GPU connections)
- 컨볼루션 층 큰 보폭으로 다운샘플링



4.4.1 AlexNet

■ AlexNet이 학습에 성공한 요인

■ 외적 요인

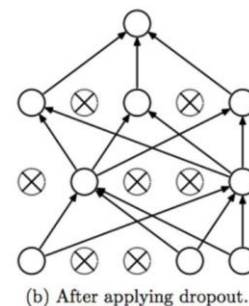
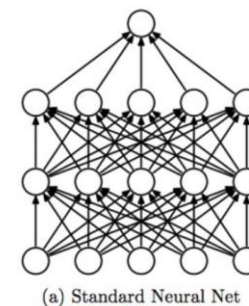
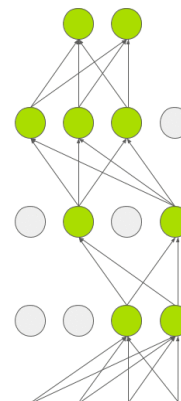
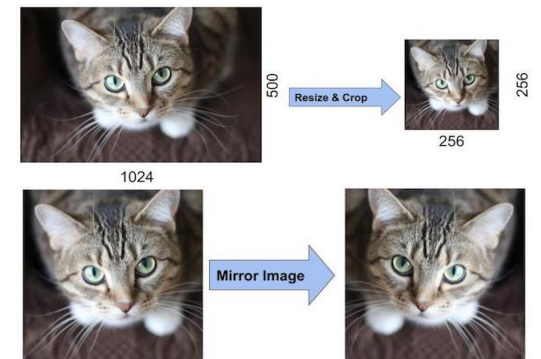
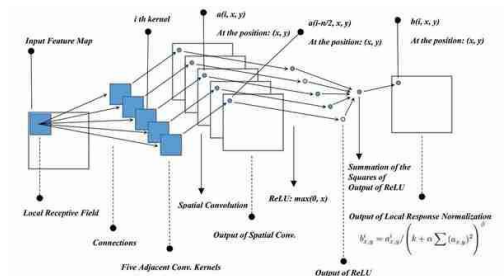
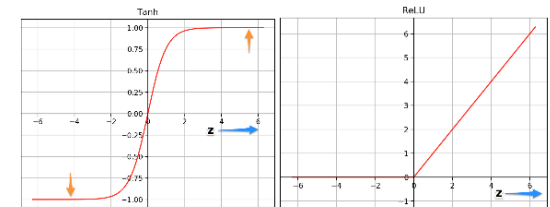
- ImageNet이라는 대규모 사진 데이터
- GPU를 사용한 병렬처리

■ 내적 요인

- 활성화함수로 ReLU 사용
- 지역 반응 정규화^{local response normalization} 기법 적용
 - 인간 신경망 측면억제^{lateral inhibition} 모방, ReLU 활성화 규제
 - 1번째, 3번째 최대 풀링 전 적용
- 과잉적합 방지하는 여러 규제 기법 적용
 - 데이터 확대 (잘라내기^{cropping}와 반전^{mirroring}으로 2048배로 확대)
 - 드롭아웃^{dropout} (완전연결층에서 사용)

■ 테스트 단계에서 앙상블 적용

- 입력된 영상을 잘라내기와 반전을 통해 증가하고 증가된 영상들의 예측 평균으로 최종 인식
- 2~3%만큼 오류율 감소 효과



4.4.2 VGGNet

■ VGGNet의 핵심 아이디어

- 3*3의 작은 커널을 사용
- 신경망을 더욱 깊게 만듦 (신경망의 깊이가 어떤 영향을 주는지 확인)
- 컨볼루션층 8~16개를 두어 AlexNet의 5개에 비해 2~3배 깊어짐

■ 16층짜리 VGG-16 (CONV 13층+FC 3층) [그림 4-22]

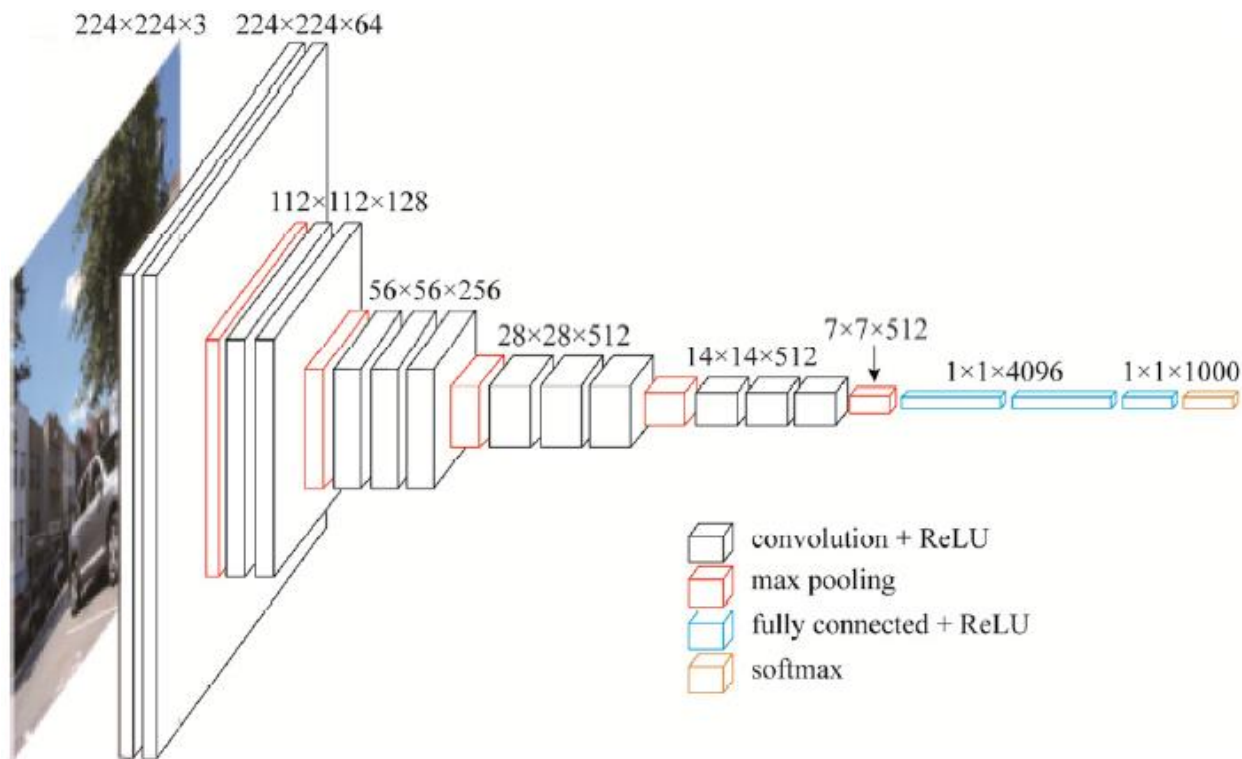
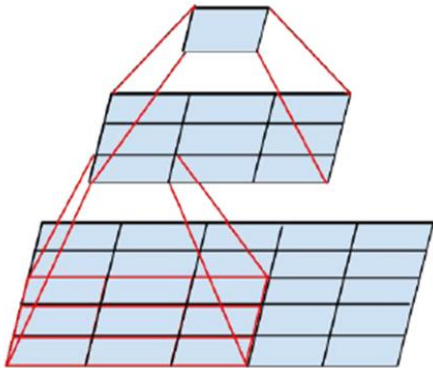


그림 4-22 VGGNet 구조[Simonyan2015]

4.4.2 VGGNet

■ 작은 커널의 이점

- GoogLeNet의 인셉션 모듈처럼 이후 깊은 신경망 구조에 영향
- 큰 크기의 커널은 여러 개의 작은 크기 커널로 분해될 수 있음
- 매개변수의 수는 줄어들면서 신경망은 깊어지는 효과
- 예) 5×5 커널을 2층의 3×3 커널로 분해하여 구현

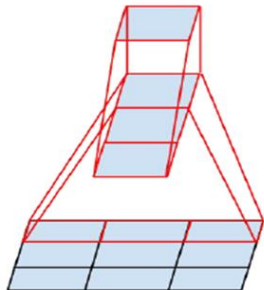


매개변수

5×5 커널인 경우, 25

3×3 커널인 경우, $9 + 9 = 18$

- 예) 3×3 커널을 1×3 커널과 3×1 커널로 분해하여 구현



매개변수

3×3 커널인 경우, 9

1×3 커널, 3×1 인 경우, $3 + 3 = 6$

→ 유사하게 $n \times n$ 커널은 $1 \times n$ 커널과 $n \times 1$ 커널로 분해될 수 있으며, n 이 클수록 매개변수의 수는 줄어드는 효과가 큼

4.4.2 VGGNet

■ 1*1 커널

- 『신경망 속의 신경망 (network in network (NIN))』에서 유래 [Lin2014]
 - VGGNet은 적용 실험을 하였지만 최종 선택하지는 않음 (GoogLeNet에서 사용됨)

- 차원 통합

■ 차원 축소 효과

- $c2 > c3$: 차원 축소 (연산량 감소)

- [그림 4-23]의 예

- $m*n$ 의 특징 맵 8개에 1*1 커널을 4개 적용 \rightarrow $m*n$ 의 특징 맵 4개 출력
- 즉, $8*m*n$ 텐서에 $8*1*1$ 커널을 4개 적용하여 $4*m*n$ 텐서를 출력하는 셈

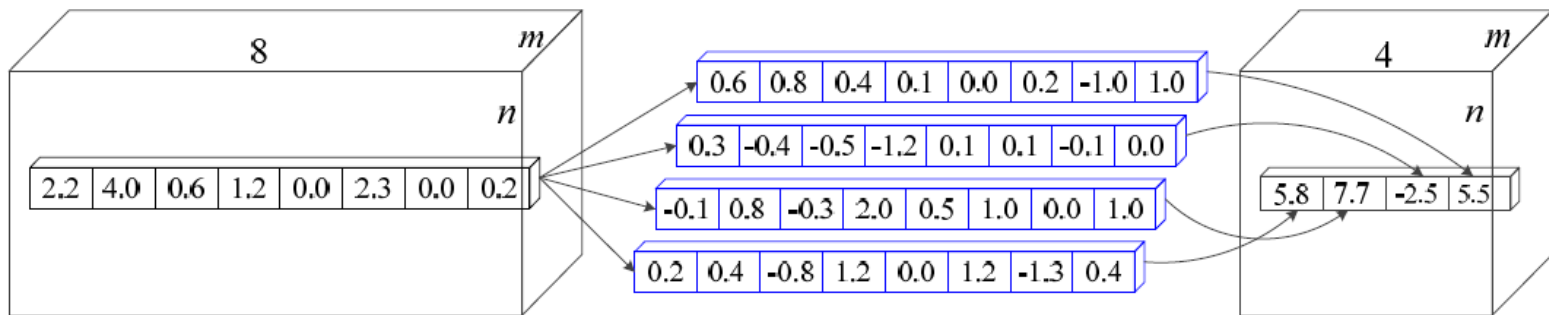
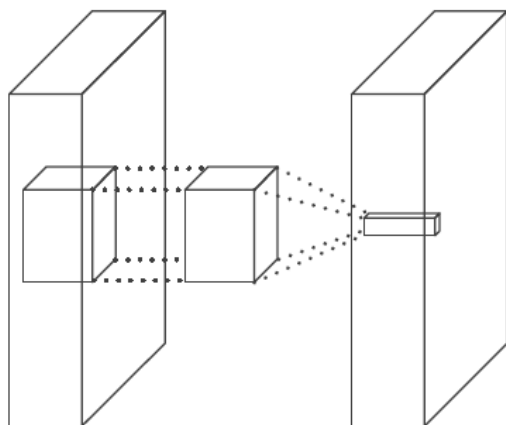


그림 4-23 1*1 컨볼루션 예제

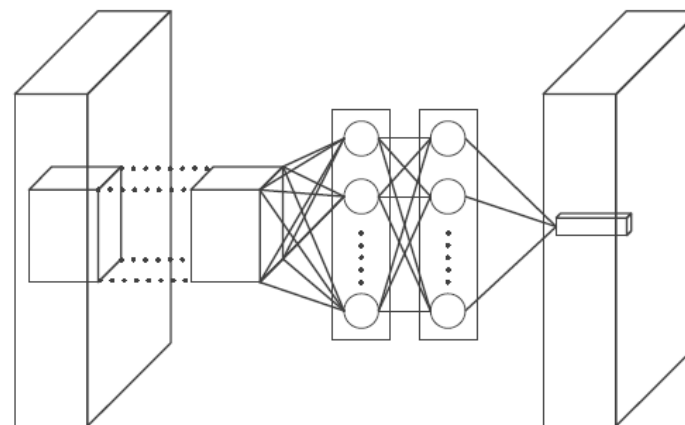
4.4.3 GoogLeNet

■ GoogLeNet의 핵심은 인셉션 모듈 inception module

- 수용장의 다양한 특징을 추출하기 위해 NIN의 구조를 확장하여 복수의 병렬적인 컨볼루션 층을 가짐
- [참고] NIN 구조
 - 기존 컨볼루션 연산을 MLPConv 연산으로 대체
 - 커널 대신 비선형 함수를 활성화함수로 포함하는 MLP를 사용하여 특징 추출 유리함
 - 신경망의 미소 신경망 micro neural network가 주어진 수용장의 특징을 추상화 시도
 - 전역 평균 풀링 global average pooling 사용



(a) 기존 컨볼루션층

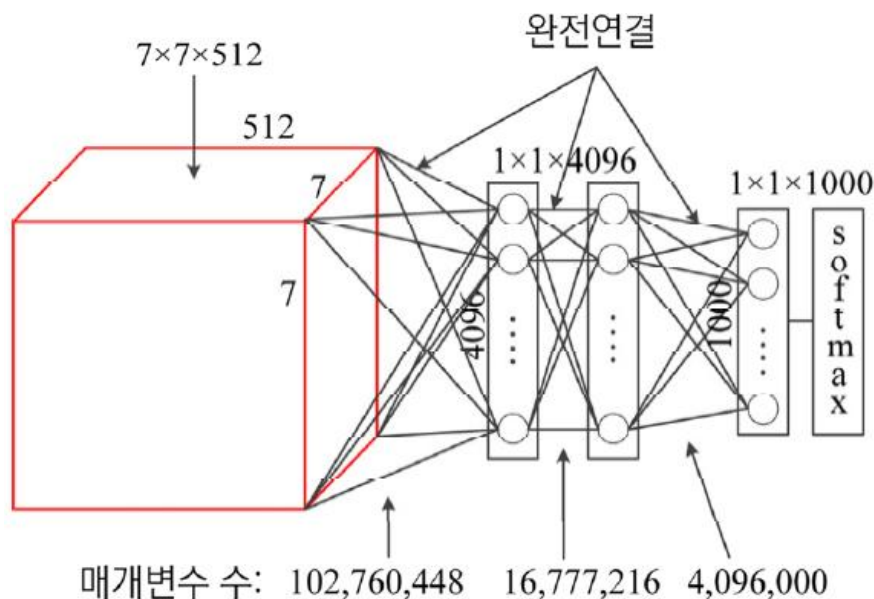


(b) NIN의 MLPconv층

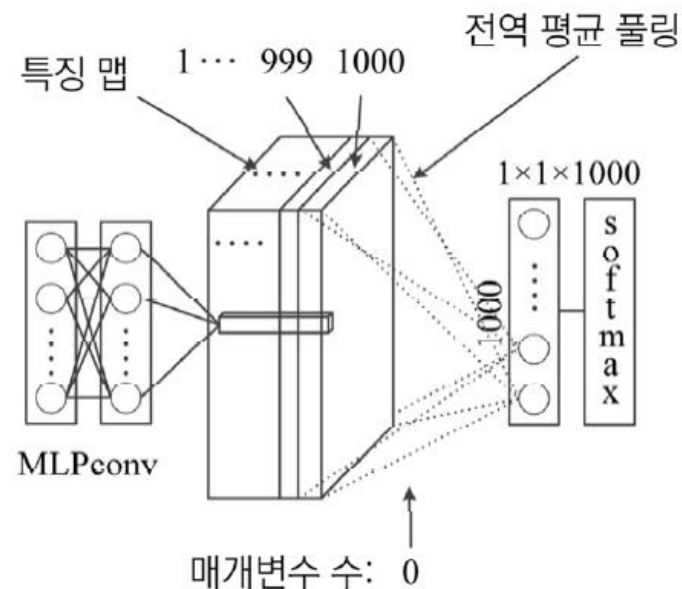
그림 4-24 기존 컨볼루션 신경망과 NIN의 비교

4.4.3 GoogLeNet

- NIN이 사용하는 전역 평균 풀링
 - [그림 4-25(a)]의 VGGNet의 완전 연결
 - » 1억2천2백만 개의 매개변수를 가짐 (VGGNet의 전체 매개변수의 85%) → 과잉적합 원인
 - [그림 4-25(b)]의 전역 평균 풀링
 - » Mlpconv가 부류 수만큼 특징 맵을 생성하면, 특징 맵 각각을 평균하여 출력 노드에 입력
 - 이 방식으로 매개변수를 줄임



(a) VGGNet의 완전연결



(b) NIN의 전역 평균 풀링

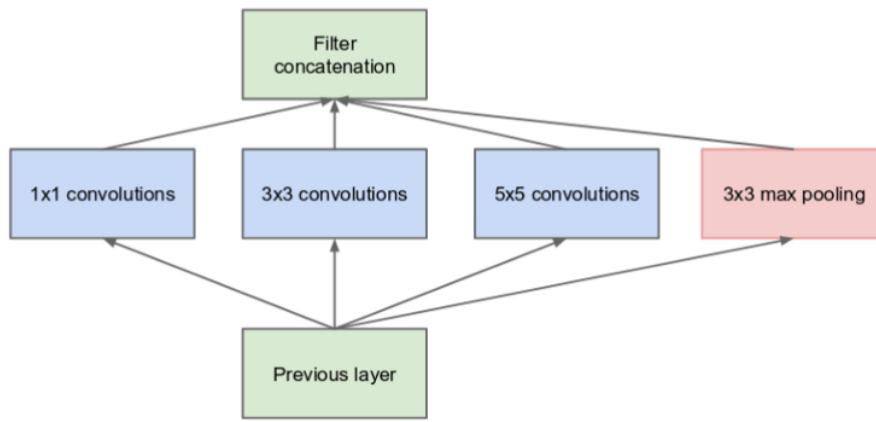
그림 4-25 완전연결과 NIN의 전역 평균 풀링의 비교

4.4.3 GoogLeNet

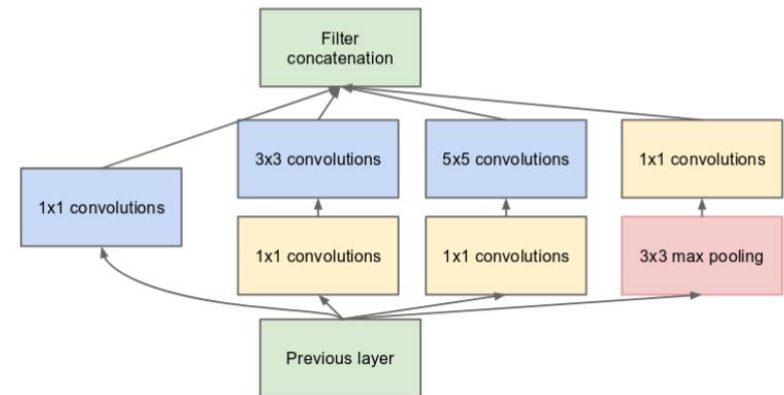
■ GoogLeNet은 NIN 개념을 확장한 신경망

■ 인셉션 모듈

- 마이크로 네트워크로 Mlpconv 대신 네 종류의 컨볼루션 연산 사용 → 다양한 특징 추출



(a) Inception module, naïve version



(b) Inception module with dimension reductions

- 1*1 컨볼루션을 사용하여 차원 축소 dimension reduction
 - 매개변수의 수 (=특징 맵의 수)를 줄임 + 깊은 신경망
- 3*3, 5*5 같은 다양한 크기의 컨볼루션을 통해서 다양한 특징들을 추출

4.4.3 GoogLeNet

■ 인셉션 모듈(I)을 9개 결합한 GoogLeNet ([그림 4-27])

- 매개변수가 있는 층 22개, 없는 층 (풀링) 5개로 총 27개 층
- 완전 연결층은 1개에 불과
 - 1백만 개의 매개변수를 가지며, VGGNet의 완전 연결층에 비하면 1%에 불과

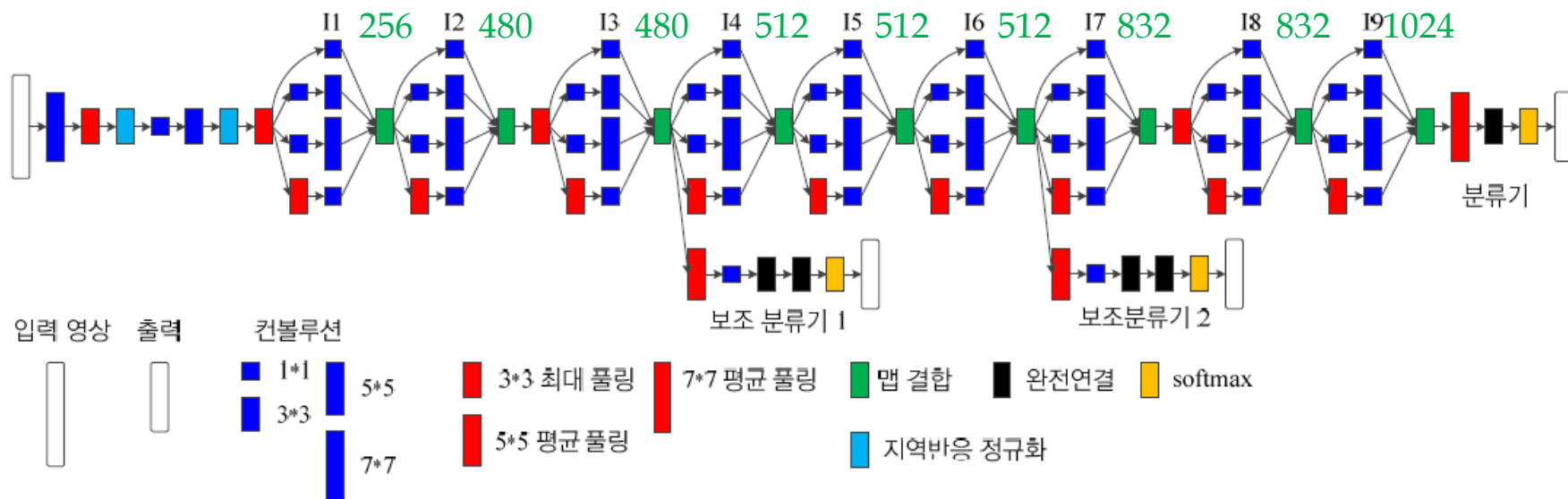


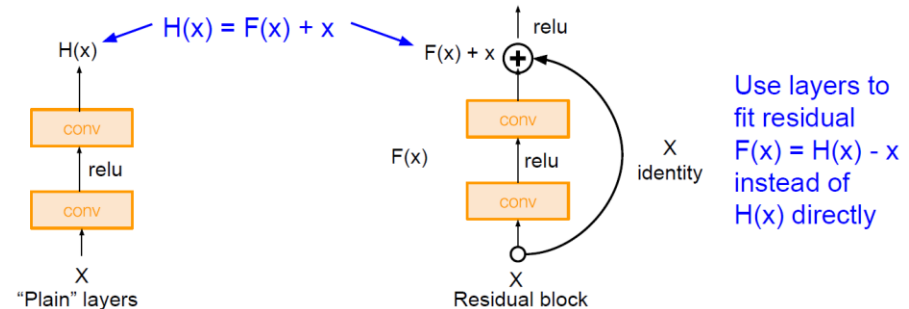
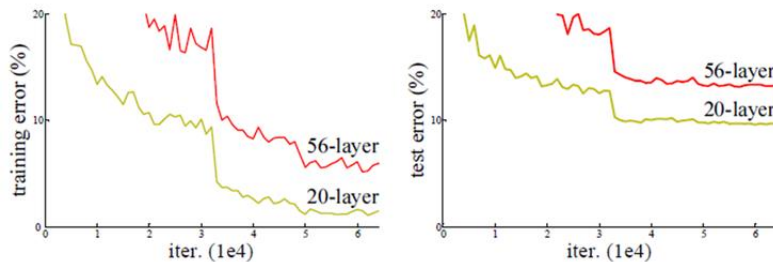
그림 4-27 GoogLeNet의 구조

- 보조 분류기 (auxiliary classifier)
 - 원 분류기의 오류 역전파 결과와 보조 분류기의 오류 역전파 결과를 결합하여 경사 소멸 문제 완화
 - 학습할 때 도우미 역할을 하고, 추론할 때 제거됨

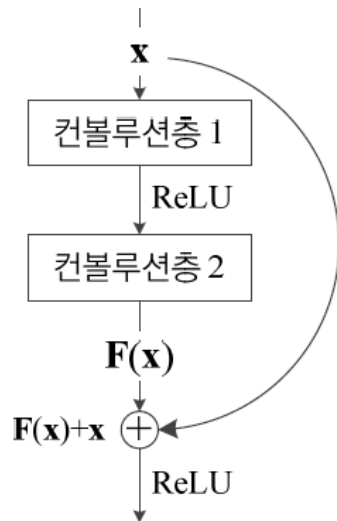
4.4.4 ResNet

■ ResNet

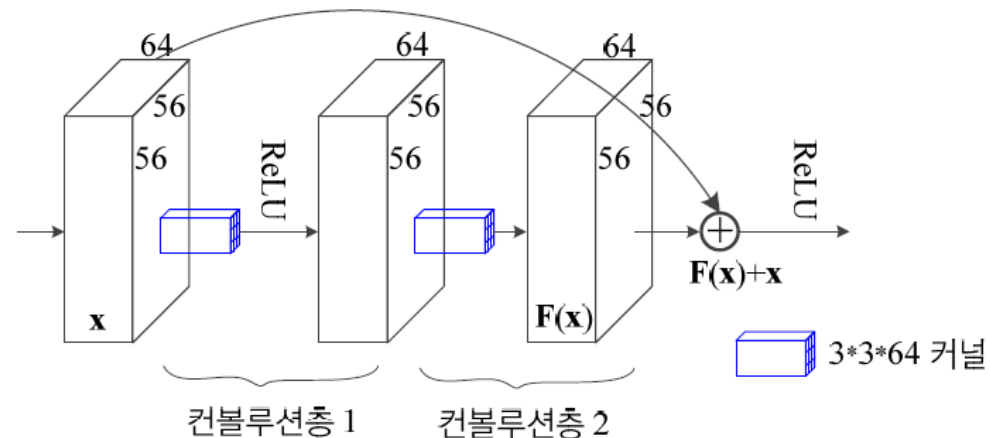
- 잔류(잔차) 학습 residual learning이라는 개념을 이용하여 성능 저하를 피하면서 **층 수를 대폭 늘림**
- 원래 컨볼루션 신경망



- 잔류 학습은 지름길 연결된 x 를 더한 $F(x) + x$ 에 τ 를 적용. $F(x)$ 는 잔류(잔차) residual $y = \tau(F(x) + x)$



(a) 빌딩블록



(b) 빌딩블록 사례

그림 4-28 잔류 학습의 구조와 동작

4.4.4 ResNet

■ 지름길 연결을 두는 이유는?

- 깊은 신경망도 최적화가 가능해짐

- 단순한 학습의 관점의 변화를 통한 신경망 구조 변화

- 단순 구조의 변경으로 매개변수 수에 영향이 없음

- 덧셈 연산만 증가하므로 전체 연산량 증가도 미비함

- 깊어진 신경망으로 인해 정확도 개선 가능함

- 경사 소멸 문제 해결

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

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

4.4.4 ResNet

■ ILSVRC 대회 성적

- 2012년 AlexNet의 15.3% 오류율은 당시 경이로운 성능
- 2015년에 ResNet은 3.5% 오류율 달성

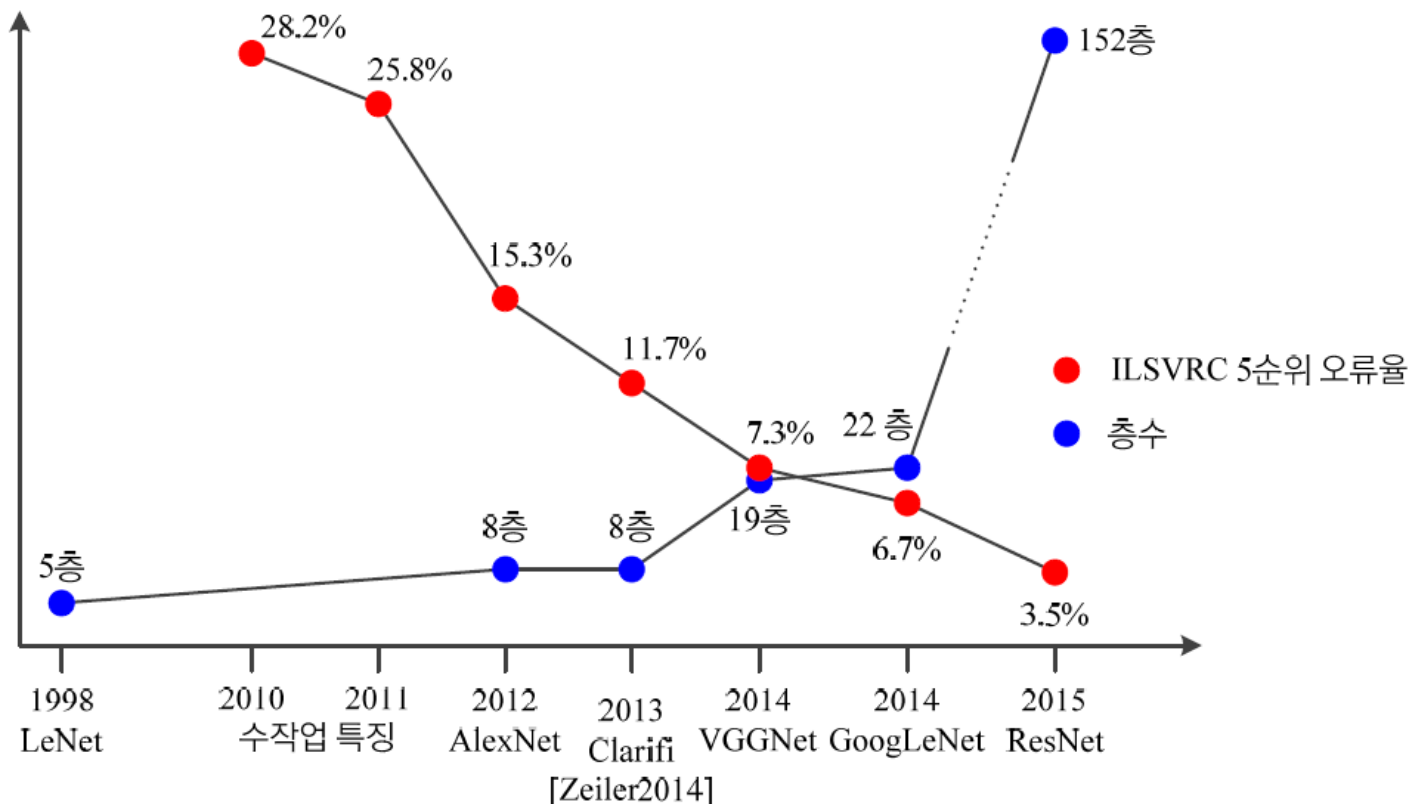


그림 4-30 CNN의 발전 추세

4.4.4 ResNet

■ ILSVRC 대회

- 분류 문제는 성능 포화 (사람 성능에 필적함)
- 물체 검출 문제에 집중함

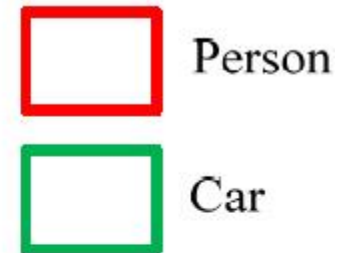


그림 4-31 ILSVRC 물체 검출 문제

4.5 생성 모델

- 4.5.1 생성 모델이란?
- 4.5.2 GAN

4.5.1 생성 모델이란?

■ 생성모델이란?

눈을 감고 황금 들녘을 상상해보라. 근사한 영상이 떠오른다면 머리 속에 있는 ‘생성 모델 generative model’이 작용한 탓이다. 사람의 생성 모델은 세상에 나타나는 현상을 오랫동안 지켜보면서 학습한 결과이다. 만일 기계 학습이 훈련집합을 사용하여 비슷한 생성 모델을 구축할 수 있다면 강한 인공지능^{strong AI}에 한발 다가설 수 있다. 왜냐하면 생성 모델은 분별 모델에 비해 데이터 생성 과정에 대한 보다 깊은 이해를 필요로 하기 때문이다[Karpathy2015].

■ 분별 모델과 생성 모델의 비교

표 4-1 분별 모델과 생성 모델

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

4.5.1 생성 모델이란?

예제 4-1

생성 모델과 분별 모델의 확률분포 추정과 예측

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

$$\mathbb{X} = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}, \mathbb{Y} = \{1, 1, 1, 0, 1, 0, 0, 0, 0, 1\}$$

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 |

4.5.1 생성 모델이란?

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 $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$ 에게 $[0.0, 0.2]$, $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ 에게 $(0.2, 0.5]$, $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ 에게 $(0.5, 0.6]$, $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ 에게 $(0.6, 1.0]$ 구간을 부여하자. 난수로 0.34가 나오면 $(0.2, 0.5]$ 에 속하므로 $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ 을 생성하고, 0.83이 나오면 $(0.6, 1.0]$ 에 속하므로 $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ 을 생성한다.

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

4.5.1 생성 모델이란?

■ 실제 상황에서 생성 모델

- 현실에 내재한 데이터 발생 분포 $P_{data}(\mathbf{x}) \rightarrow$ 알아낼 수 없음

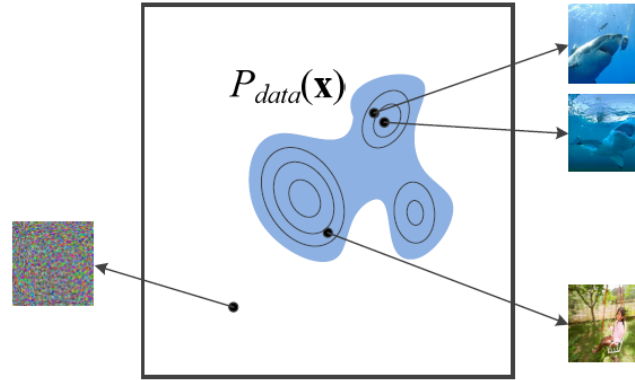
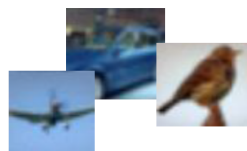


그림 4-32 확률분포 $P_{data}(\mathbf{x})$ 에 따른 자연 영상의 발생 과정

- $P_{data}(\mathbf{x})$ 를 모방하는 모델의 확률 분포 $P_{model}(\mathbf{x}; \Theta)$
 - $P_{model}(\mathbf{x}; \Theta)$ 를 명시적으로 추정하는 것도 어려움
 - 현대 기계 학습은 주로 딥러닝 모델을 사용하여 확률 분포를 암시적으로 표현
 - GAN (generative adversarial network), VAE (variational autoencoder), RBM (10장)

Given training data, generate new samples from same distribution



Training data $\sim p_{data}(\mathbf{x})$



Generated samples $\sim p_{model}(\mathbf{x})$

Want to learn $p_{model}(\mathbf{x})$ similar to $p_{data}(\mathbf{x})$

4.5.1 생성 모델이란?

■ 생성 모델 분류

Taxonomy of Generative Models

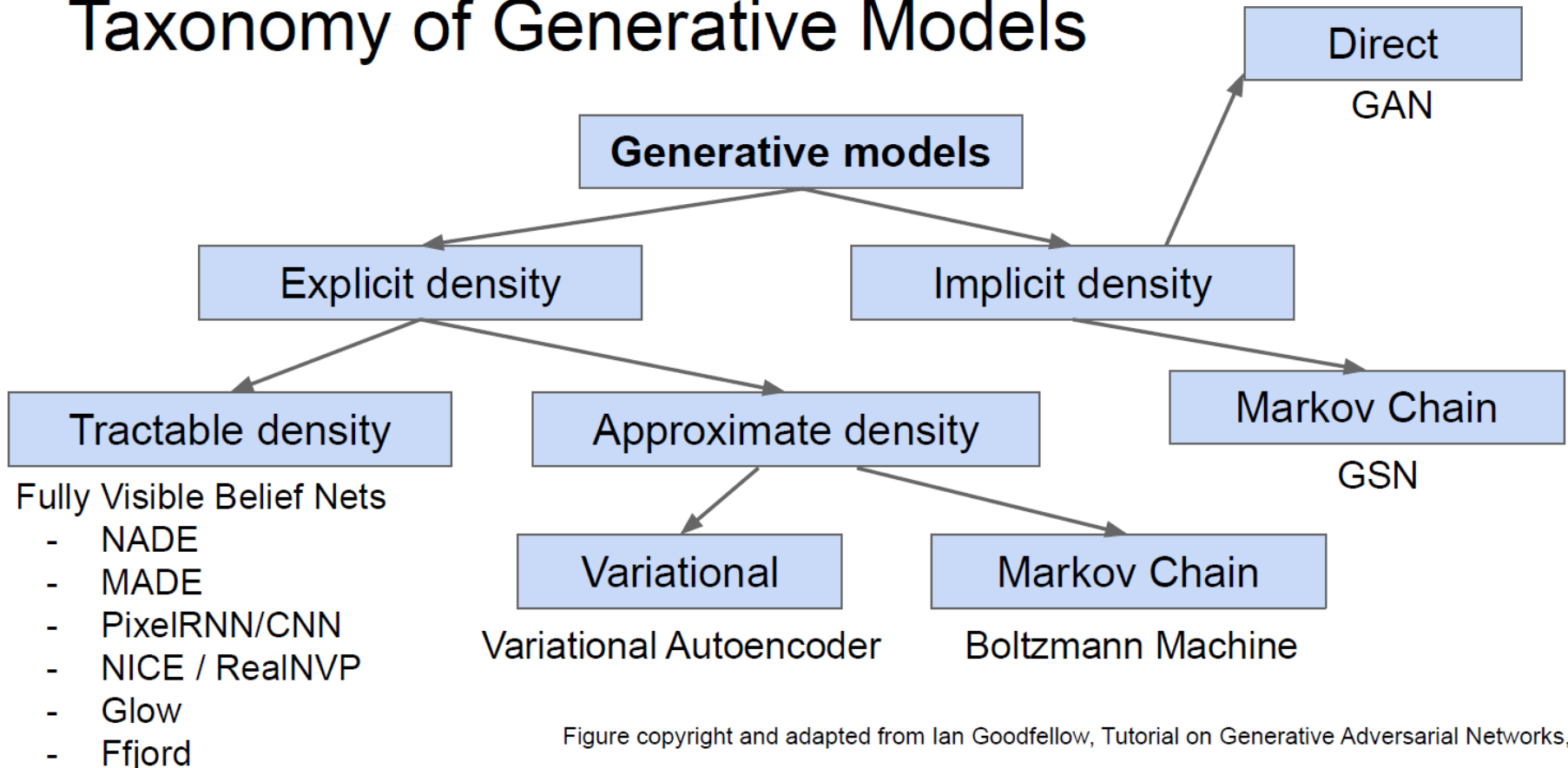


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

4.5.2 GAN(generative adversarial network)

■ GAN의 우월한 성능

- 사람을 상대로 진짜와 가짜 구별하는 실험에서
MNIST 52.4%, CIFAR-10 78.7% (50%이면 완벽히 속임)

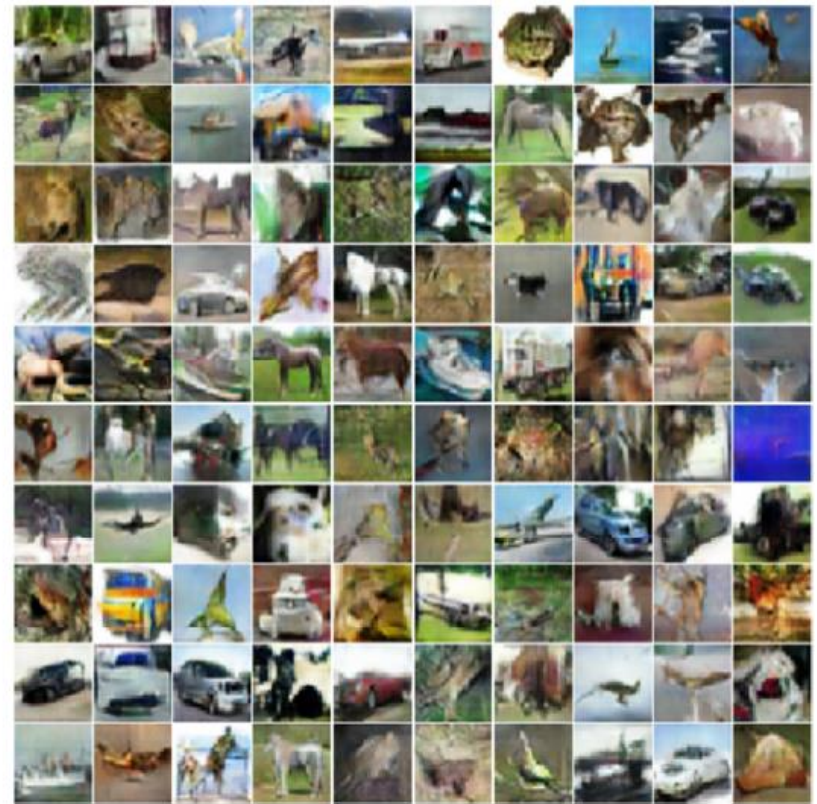


그림 4-33 GAN으로 생성한 영상 샘플

4.5.2 GAN

■ GAN의 핵심

- 생성기(generator G)와 분별기(discriminator D)의 대립 구도

- G는 가짜 샘플 생성 (위조지폐범)
- D는 가짜와 진짜를 구별 (경찰)

- GAN의 목표는 위조지폐범의 승리

(G가 만들어내는 샘플을 D가 구별하지 못하는 수준까지 학습)

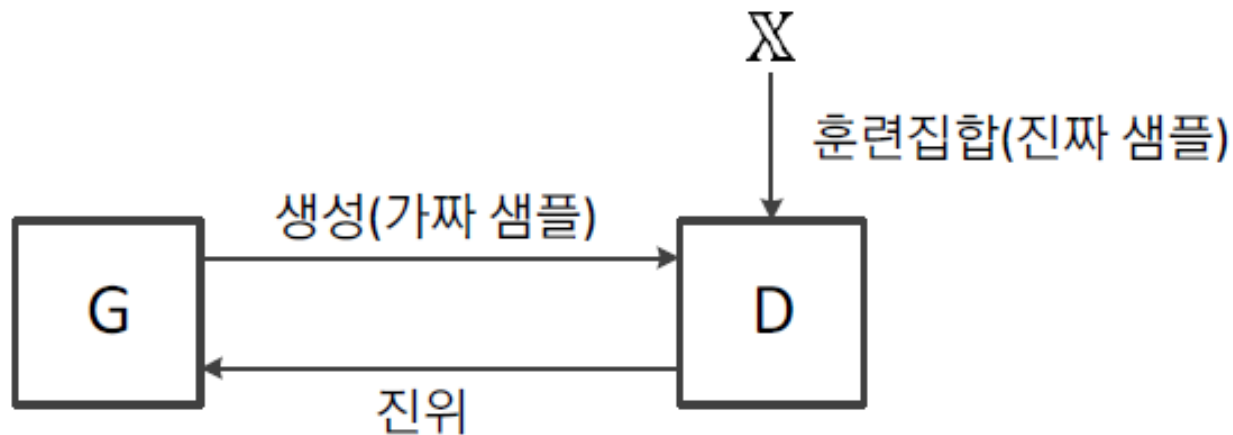


그림 4-34 GAN의 원리