

5장. 딥러닝 - III

3. 딥러닝 생성 모델

3.1 제한적 볼츠만 머신 RBM

3.2 심층 신뢰망 DBN

3.3 대립쌍 생성망 GAN

3. 딥러닝 생성 모델

❖ 기계학습 모델

▪ 판별 모델(discriminative model)

- 분류나 회귀 문제에서 입력이 주어지면 출력값을 직접 결정하는 함수나 패턴
- 확률분포로 표현한 모델 : $p(y|x)$
- 의사결정트리, 입출력 관계를 학습한 신경망 모델, 서포트 벡터 머신

▪ 생성 모델(generative model)

- 분류, 회귀 또는 군집화 문제 등에서 사용되는 특정 분포의 데이터가 만들어질 때 사용되는 함수나 절차
- 일반적으로 관심대상인 학습 데이터를 가장 잘 표현하는 확률분포로 표현
- 확률분포로 표현한 모델 : $p(x, y)$
- 확률 그래프 모델(제한적 볼츠만 머신, 심층 신뢰망 등), 대립쌍 생성망 (GAN)

딥러닝 생성 모델

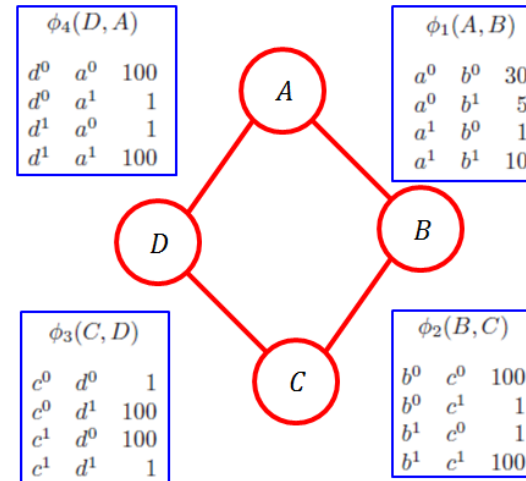
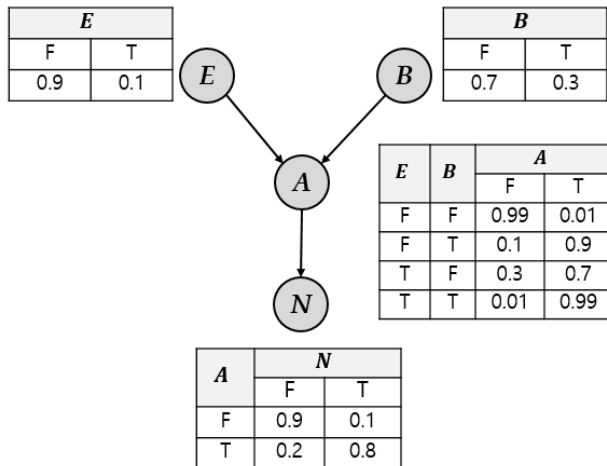
❖ 딥러닝에서 생성 모델

- 학습된 모델이 학습 데이터와 유사한 데이터를 생성할 수 있는 모델
- **확률 기반 모델**
 - 제한적 볼츠만 머신(RBM) 기반 **심층신뢰망**(Deep Belief Network, **DBN**)
 - **딥볼츠만 머신**(deep Boltzmann machine, **DBM**)
- **비확률 기반 모델**
 - **대립쌍 생성망**(generative adversarial net, **GAN**)

3.1 제한적 볼츠만 머신

❖ 확률 그래프 모델(probabilistic graphical model)

- 확률변수에 대응하는 **노드(node)**와 확률변수 간의 관계를 나타내는 **간선(edge)**으로 구성된 **그래프**를 사용하여 **결합 확률분포(joint probability distribution)** 표현
- **베이지안 망(Bayesian network)**
 - 조건부 확률을 사용하여 결합 확률 분포 표현
- **마르코프 랜덤 필드(Markov random field)**
 - 호응정도(affinity) 표나 팩터(factor)를 사용하여 결합확률 분포 표현



제한적 볼츠만 머신

❖ 에너지 개념을 이용한 확률 표현

▪ 통계 물리학

- 동일한 상태에 있는 물질이라도 다양한 미시적 상태(micro state)에 있을 수 있음
- 높은 에너지의 미시적 상태의 있을 확률은 낮음
- 낮은 에너지의 미시적 상태에 있을 확률은 높음

▪ 확률 그래프 모델

- 물질이 특정 형상(configuration)의 상태에 있을 확률 표현 가능
 - 형상: 노드(확률변수)들에 허용되는 값의 조합
 - 특정 형상에 해당하는 미시적 상태에 대한 에너지 정의
 - 형상의 집합 : $\{s_1, s_2, \dots, s_N\}$
 - 에너지의 집합 : $\{\epsilon_1, \epsilon_2, \dots, \epsilon_N\}$
- 물질이 형상 s_i 에 있을 확률 $p(s_i)$ 정의

$$p(s_i) = \frac{e^{-\beta\epsilon_i}}{\sum_j e^{-\beta\epsilon_j}} = \frac{e^{-\beta\epsilon_i}}{Z} \quad Z = \sum_j e^{-\beta\epsilon_j} \quad (\text{분할 함수})$$

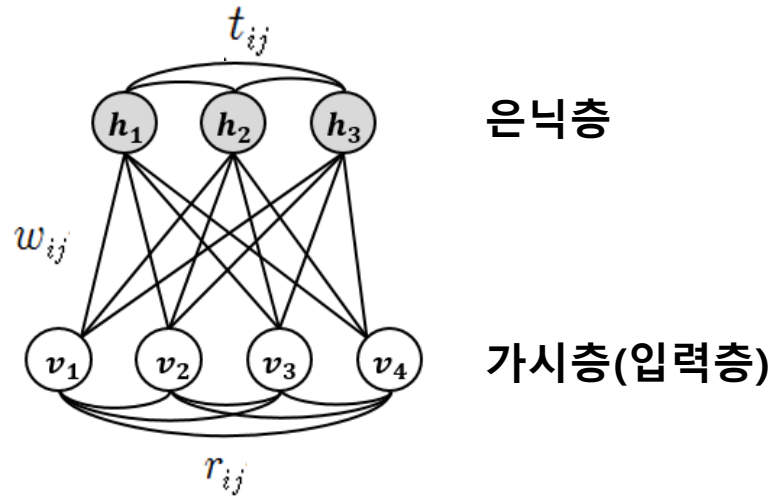
볼츠만 분포(Boltzmann distribution) 또는 깁스 분포(Gibbs distribution)

- 에너지 ϵ_i 가 줄어들 수록 확률 값은 증가

3.1.1 볼츠만 머신과 제한적 볼츠만 머신

❖ 볼츠만 머신(Boltzmann Machine)

- 마르코프 모델(Markov model)의 일종
 - 모든 노드 간에 간선이 연결된 마르코프 모델
- 확률분포가 에너지 기반 모델로 정의



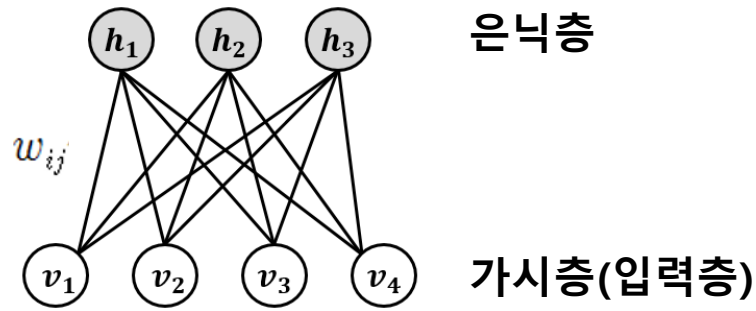
- 형상(\mathbf{v}, \mathbf{h})에 대한 에너지의 정의

$$\epsilon(\mathbf{v}, \mathbf{h}) = - \left(\sum_{i < j} r_{ij} v_i v_j + \sum_{i < j} w_{ij} v_i h_j + \sum_{i < j} t_{ij} h_i h_j + \sum_i a_i v_i + \sum_j b_j h_j \right)$$

볼츠만 머신과 제한적 볼츠만 머신

❖ 제한적 볼츠만 머신 (Restricted Boltzmann Machine, RBM)

- 가시층 노드와 은닉층 간에 간선이 없는 볼츠만 머신
- 에너지 함수



$$\epsilon(v, h) = - \left(\sum_{i < j} w_{ij} v_i h_j + \sum_i a_i v_i + \sum_j b_j h_j \right)$$

볼츠만 머신과 제한적 볼츠만 머신

❖ 확률 그래프 모델의 학습

- 확률분포를 표현하기 위해, 그래프 구조를 결정하고 에너지 함수를 정의하여, 간선의 가중치 및 노드의 편차항 등을 결정하는 것

❖ 확률 그래프 모델의 추론(Inference)

- 문제에 주어진 정보를 해당 노드에 입력값으로 설정한 다음, 관심 대상인 노드들의 확률분포를 계산해 내는 것

3.1.2 제한적 볼츠만 머신의 추론

❖ 제한적 볼츠만 머신의 추론

- 결합확률 분포

$$p(\mathbf{v}, \mathbf{h}) = \frac{e^{-\epsilon(\mathbf{v}, \mathbf{h})}}{Z} \quad Z = \sum_{\mathbf{v}, \mathbf{h}} e^{-\epsilon(\mathbf{v}, \mathbf{h})}$$

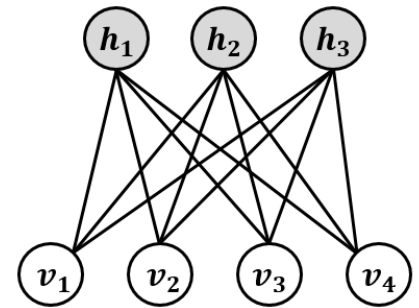
- 가시층이 데이터 \mathbf{v} 를 가질 확률 $p(\mathbf{v})$

$$p(\mathbf{v}) = \frac{1}{Z} \sum_{\mathbf{h}} e^{-\epsilon(\mathbf{v}, \mathbf{h})}$$

- 조건부 확률

$$p(\mathbf{h}|\mathbf{v}) = \prod_{j=1}^m p(h_j|\mathbf{v})$$

$$p(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^n p(v_i|\mathbf{h})$$

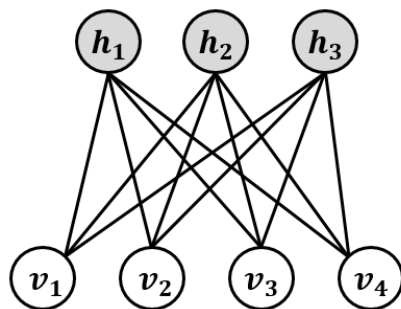


제한적 볼츠만 머신의 추론

❖ 제한적 볼츠만 머신의 추론 - cont.

- 노드(확률변수)가 이진값을 가질 때 노드의 확률

$$p(h_j = 1 | \mathbf{v}) = \frac{1}{1 + \exp\left(-b_j - \sum_i v_i w_{ij}\right)} = \text{sigm}\left(b_j + \sum_i v_i w_{ij}\right)$$



$$p(v_i = 1 | \mathbf{h}) = \frac{1}{1 + \exp\left(-a_i - \sum_j h_j w_{ij}\right)} = \text{sigm}\left(a_i + \sum_j h_j w_{ij}\right)$$

3.1.3 제한적 볼츠만 머신의 학습

❖ 제한적 볼츠만 머신의 학습

▪ 목적함수

- 모든 학습 데이터에 대해 높은 확률값을 갖도록 가중치 w_{ij} 와 편차항 a_i, b_j 를 결정

- **학습 데이터:** 가시층에 입력으로 제공되는 데이터

$$V = \{v_1, v_2, \dots, v_M\}$$

- **가능도(likelihood)**

- RBM와 같은 확률 모델을 사용하여 계산한 데이터에 대한 확률

- **전체 학습 데이터에 대한 가능도**

- 각 학습 데이터의 가능도를 곱한 것

$$p(V) = \prod_{i=1}^M p(v_i)$$

- **로그 가능도(log likelihood)** \Rightarrow 주로 사용하는 목적함수

$$l = \sum_{i=1}^M \log p(v_i)$$

제한적 볼츠만 머신의 학습

❖ 제한적 볼츠만 머신의 학습 - cont.

- 최대 가능도 추정(maximum likelihood estimation) 적용
 - 전체 학습 데이터에 대한 가능도를 최대로 만드는 파라미터 선택
 - 경사 상승법(gradient-ascent method) 사용

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \eta \frac{\partial l}{\partial w_{ij}}$$

$$a_i^{(t+1)} = a_i^{(t)} + \eta \frac{\partial l}{\partial a_i}$$

$$b_j^{(t+1)} = b_j^{(t)} + \eta \frac{\partial l}{\partial b_j}$$

제한적 볼츠만 머신의 학습

❖ 제한적 볼츠만 머신의 학습 - cont.

- 경사 상승법 (gradient-ascent method)을 위한 편미분

$$\frac{\partial l_v}{\partial w_{ij}} = \frac{\partial \log p(\mathbf{v})}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}$$

$$\langle v_i h_j \rangle_{data} = p(h_j = 1 | \mathbf{v}) v_i$$

학습 데이터로 부터 쉽게 계산 가능

$$\langle v_i h_j \rangle_{model} = \sum_{\mathbf{v}} p(\mathbf{v}) p(h_j = 1 | \mathbf{v}) v_i$$

가시층 노드에 대한 확률분포를 고려해야 하므로 직접적인 계산 곤란

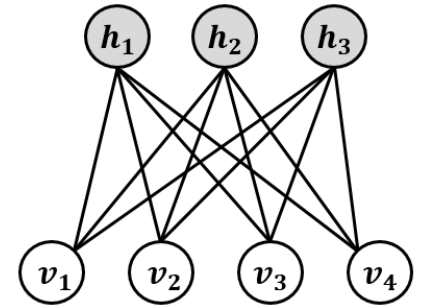
$$\frac{\partial l_v}{\partial a_i} = \langle v_i \rangle_{data} - \langle v_i \rangle_{model}$$

$$\frac{\partial l_v}{\partial b_j} = \langle h_j \rangle_{data} - \langle h_j \rangle_{model}$$

제한적 볼츠만 머신의 학습

❖ $\langle v_i h_j \rangle_{model}$ 의 계산

- 마르코프 체인 몬테카를로(Markov Chain Monte Carlo, MCMC) 사용
- RBM은 마르코프 체인의 일종
- 몬테카를로 기법(Monte Carlo method)
 - 주어진 확률 분포로 많은 표본 추출
 - 표본을 사용하여 기대값 추정



❖ 깃스 표본추출(Gibbs sampling)

- 다른 모든 확률변수의 값은 고정되어 있는 것으로 간주한 상태에서 어떤 확률변수의 값을 표본추출

$$h_1 \sim p(h_1 | v_1, v_2, v_3, v_4)$$

$$h_2 \sim p(h_2 | v_1, v_2, v_3, v_4)$$

$$h_3 \sim p(h_3 | v_1, v_2, v_3, v_4)$$

$$v_1 \sim p(v_1 | h_1, h_2, h_3)$$

$$v_2 \sim p(v_2 | h_1, h_2, h_3)$$

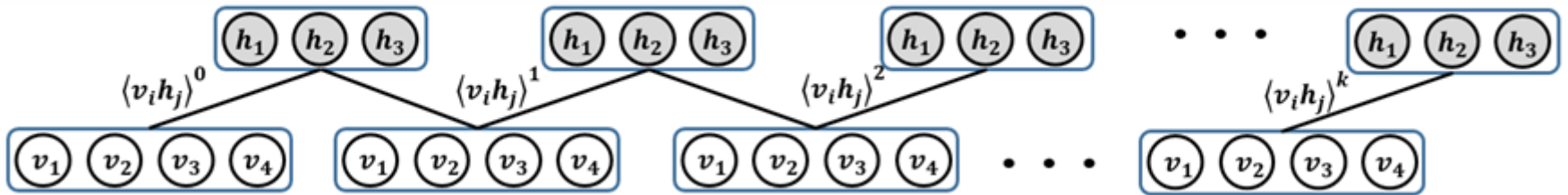
$$v_3 \sim p(v_3 | h_1, h_2, h_3)$$

$$v_4 \sim p(v_4 | h_1, h_2, h_3)$$

제한적 볼츠만 머신의 학습

❖ k -단계 대조분기(k -step contrastive divergence)

- 학습 데이터 v 에 대한 k 번 깃스 표본추출을 반복한 다음, 마지막 단계의 가시 노드와 은닉노드의 값 $\langle v_i^k h_j^k \rangle$ 을 $\langle v_i h_j \rangle_{model}$ 대신 사용



$$\frac{\partial l_v}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}$$

$$\frac{\partial l_v}{\partial a_i} = \langle v_i \rangle_{data} - \langle v_i \rangle_{model}$$

$$\frac{\partial l_v}{\partial b_j} = \langle h_j \rangle_{data} - \langle h_j \rangle_{model}$$

제한적 볼츠만 머신의 학습

알고리즘 5.1 k -단계 대조분기 기반의 RBM 학습 알고리즘

입력 : 학습 데이터 집합 $V = \{(v_1, v_2, \dots, v_M)\}$, 학습률 η

출력 : 가중치 벡터 W , 편차항 벡터 a, b

1. W, a, b 를 작은 무작위 값으로 초기화한다.

2. repeat

3. for (V 의 각 학습 데이터 v 에 대해) {

4. $v^0 \leftarrow v$

5. for ($t = 1$ to k) {

6. (식 5.57)를 이용하여 $p(h_j^{k-1} = 1 | v^{k-1})$ 를 계산하고 h^{k-1} 표본추출

7. (식 5.58)을 이용하여 $p(v_i^k = 1 | h^{k-1})$ 를 계산하고 v^k 표본추출

8. }

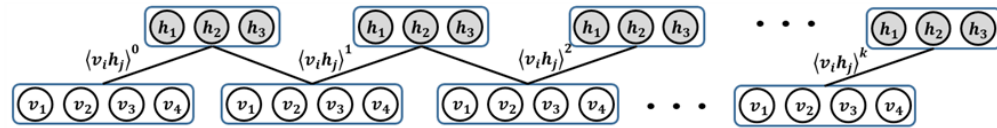
9. (식 5.68)을 이용하여 노드간 연결 가중치 w_{ij} 수정

10. (식 5.72)를 이용하여 가시 노드의 편차항 a_i 수정

11. (식 5.73)을 이용하여 은닉 노드의 편차항 b_j 수정

12. }

13. until (종료조건 만족)



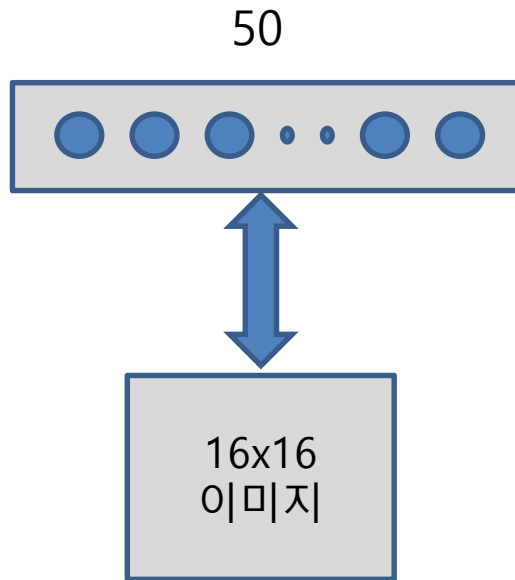
$$\frac{\partial l_v}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}$$

$$\frac{\partial l_v}{\partial a_i} = \langle v_i \rangle_{data} - \langle v_i \rangle_{model}$$

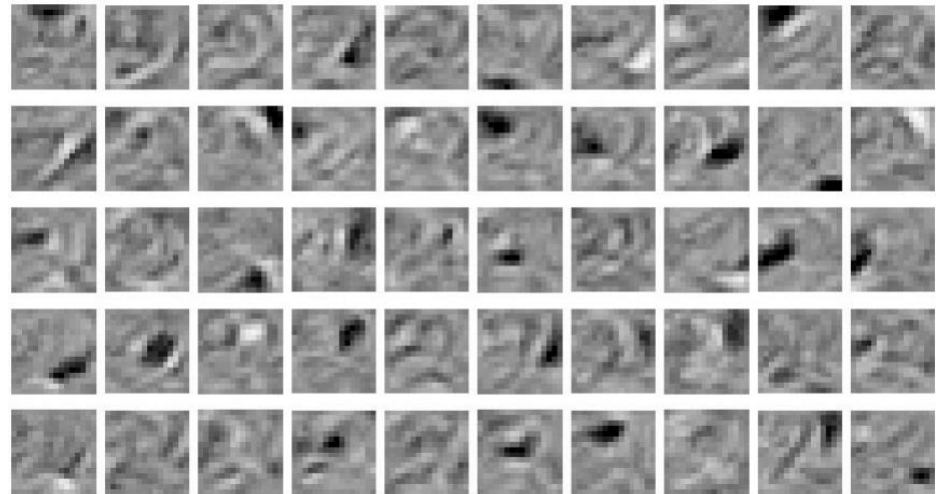
$$\frac{\partial l_v}{\partial b_j} = \langle h_j \rangle_{data} - \langle h_j \rangle_{model}$$

제한적 볼츠만 머신의 학습

❖ MNIST 학습 데이터에 대한 RBM의 학습 예



최종 50 x 256 가중치

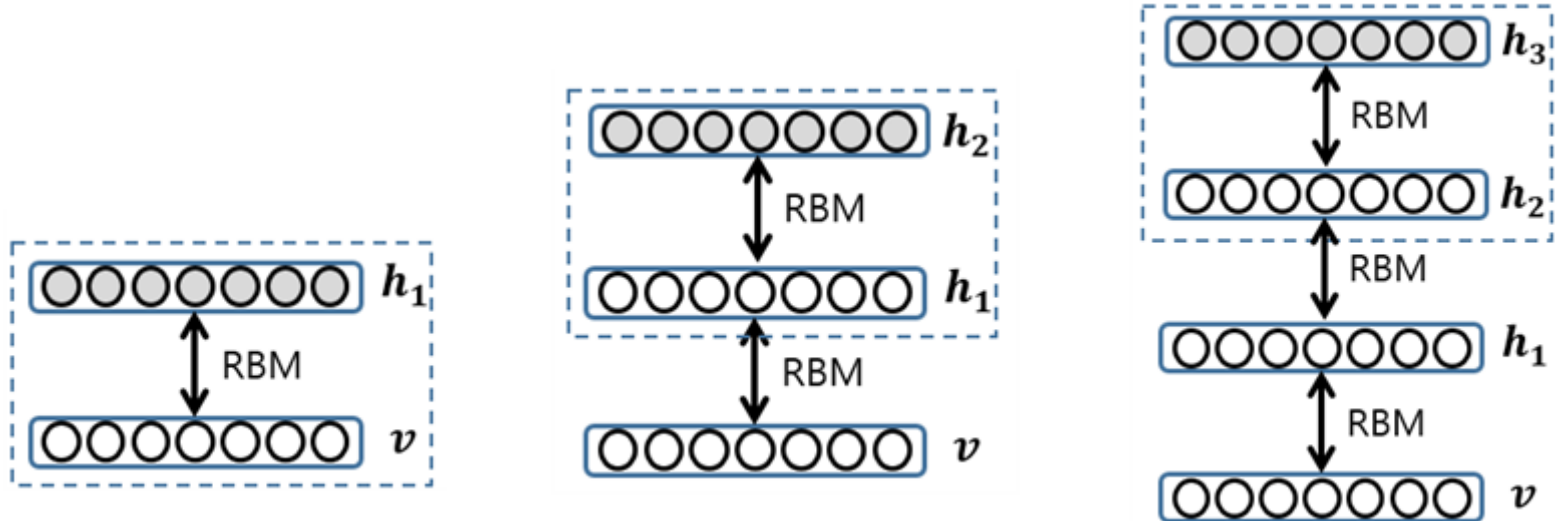


학습데이터 재구성 데이터

3.1.4 제한적 볼츠만 머신을 이용한 사전 학습

❖ 적층 제한적 볼츠만 머신(Stacked RBM)

- 순차적으로 RBM을 학습시켜가면서 계층적 구조 생성
- 비지도 학습으로 학습
- 위로 올라갈 수록 더 추상적인 특징 추출
- 학습된 가중치를 다층 퍼셉트론(MLP)의 가중치 초기값을 사용



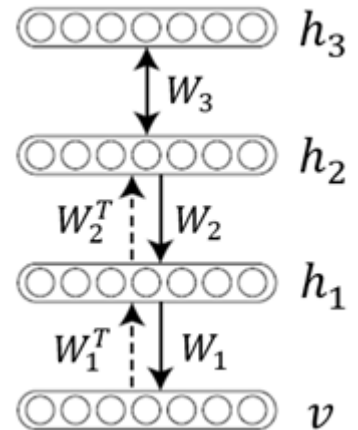
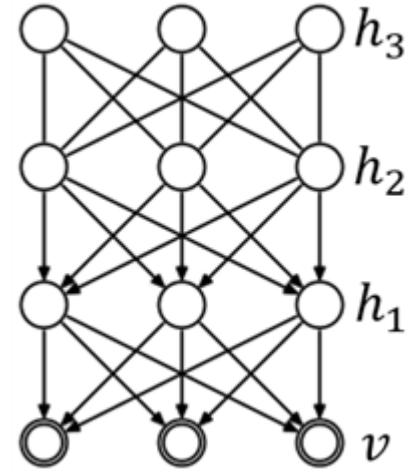
3.2 심층 신뢰망(DBN)

❖ 심층 신뢰망(Deep Belief Network, DBN)

- RBM와 방향성 그래프 모델의 결합 형태
- **가시층에서 학습 데이터의 가능도가 최대화** 되도록 학습
- **생성 모델의 역할** 보유
 - 최상위 층에 무작위 값을 설정하면 가시층에 값 생성

▪ 학습 방법

1. 적층 RBM 형태로 학습하여 가중치 결정
2. **wake-sleep** 알고리즘을 적용하여 가중치 상세 조정
 - 위로 올라가는 가중치 W_1^T, W_2^2 추가
 - wake와 sleep 단계 반복
 - **wake 단계**
 - » W_1^T, W_2^2 를 사용하여 위로 가는 방향의 노드값 결정
 - » 아래로 가는 가중치 W_1, W_2 조정
 - **sleep 단계**
 - » 아래로 가는 방향의 노드값 결정
 - » 위로 가는 가중치 조정



심층 신뢰망(DBN)

❖ 필기체 숫자 인식 DBN의 예

- 28x28 크기의 숫자 이미지
- 10개 노드의 출력

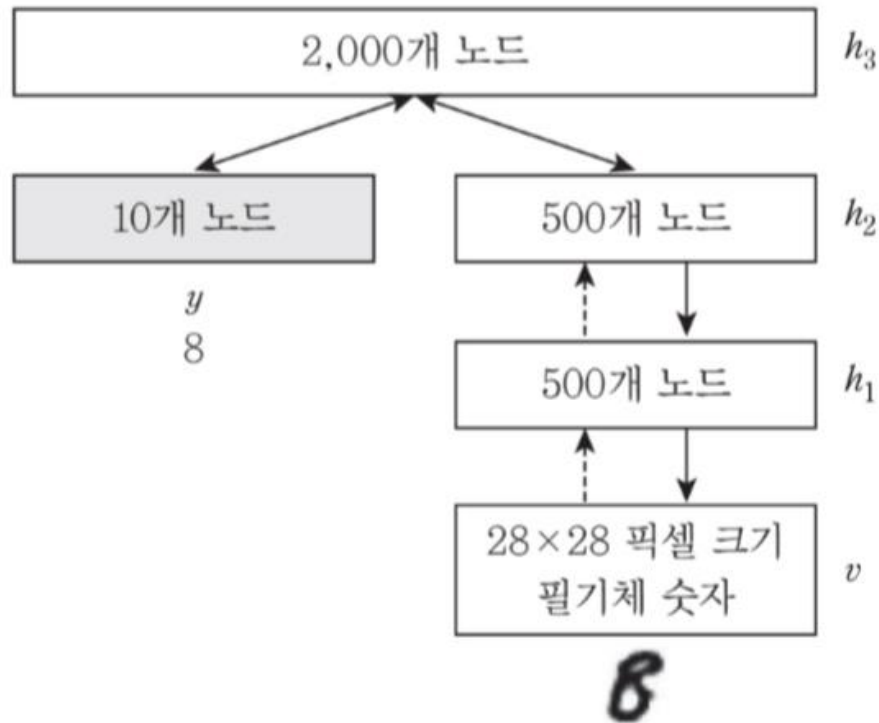


그림 5.42 필기체 숫자 인식 DBN.

심층 신뢰망(DBN)

❖ 필기체 숫자 인식 DBN의 예 - cont.

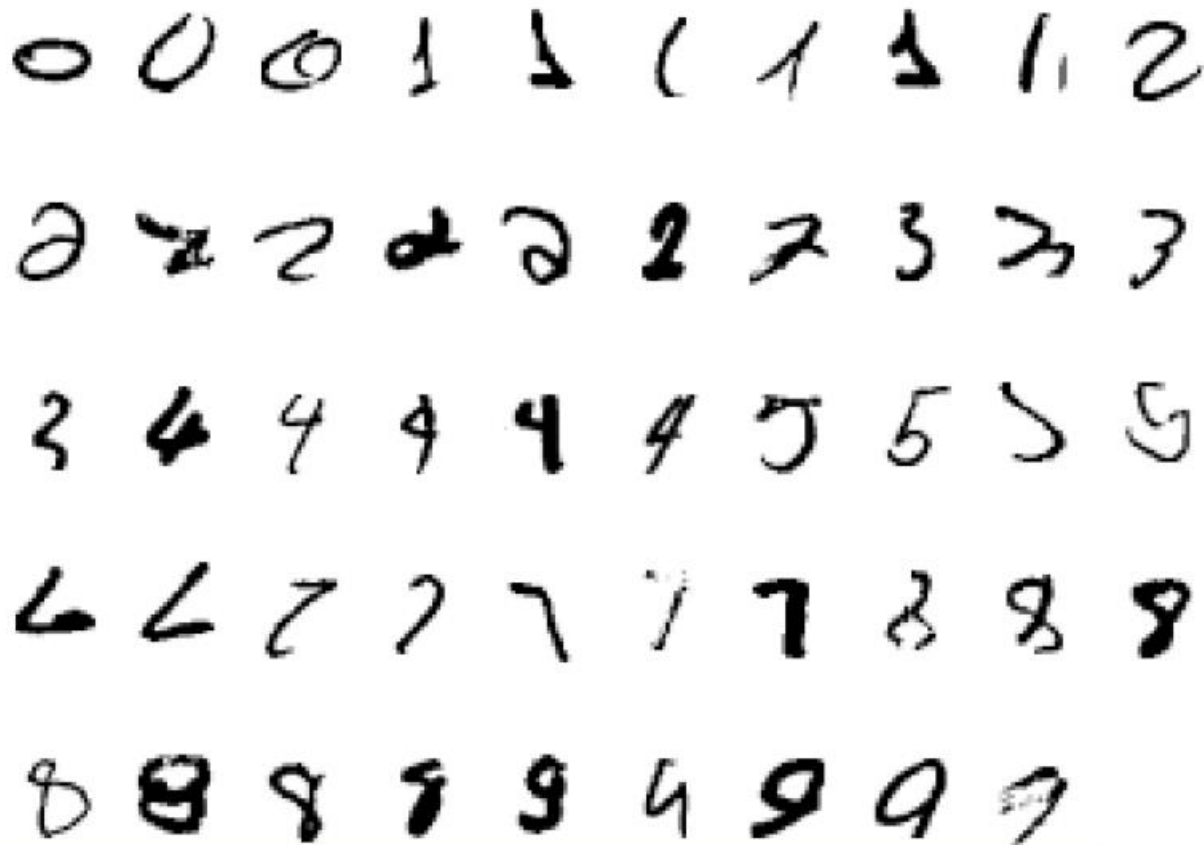
- 깃스 표본추출을 1,000번 반복한 후에 생성되는 표본들의 사례



심층 신뢰망(DBN)

❖ 필기체 숫자 인식 DBN의 예 - cont.

- 정확히 인식된 필기체 숫자의 예



심층 볼츠만 머신

- ❖ 심층 볼츠만 머신(Deep Boltzmann Machine)
 - 모든 층이 RBM의 구성
 - 층간의 사용작용 때문에 학습이 어려움

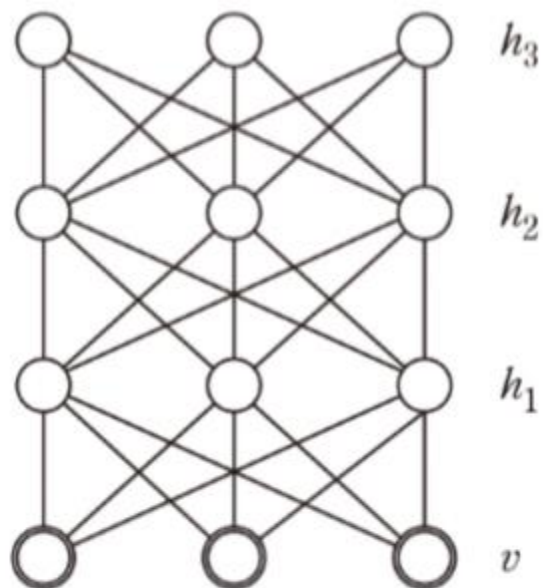


그림 5.43 심층 볼츠만 머신

3.3 대립쌍 생성망

❖ 대립쌍 생성망 (Generative Adversarial Network, GAN)

- 굿펠로우 등이 2014년에 개발
- 주어진 학습 데이터의 특성을 잘 만족하는 데이터 자체를 생성하는 모델 만드는 데 관심
- 생성 모듈(generator)과 판별 모듈(discriminator)로 구성
- 화폐 위조범과 경찰관의 관계와 학습
 - 경찰관 : 위폐 판별을 위한 학습
 - 위조범 : 경찰을 속일 만큼 정교한 위폐를 만들기 위해 학습

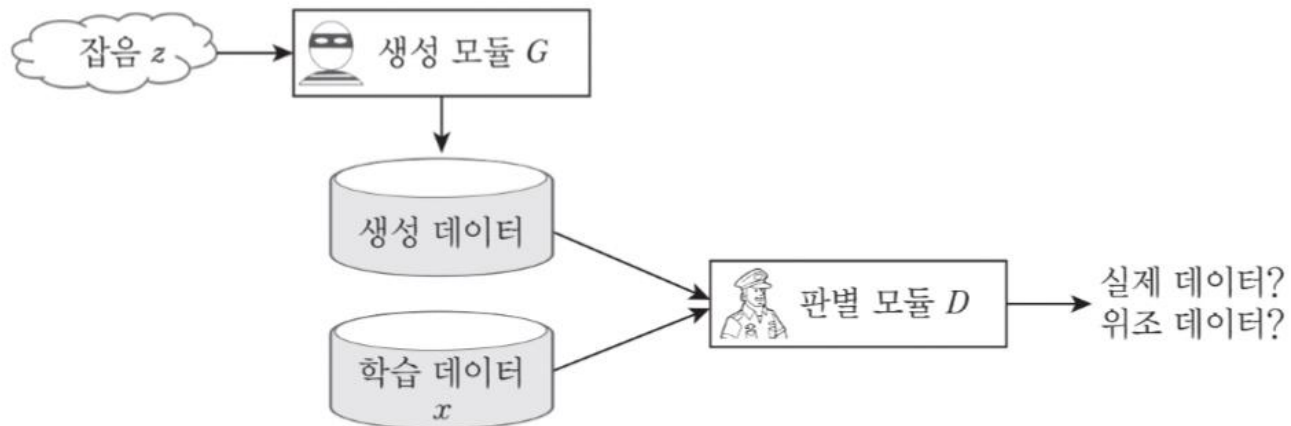


그림 5.44 대립쌍 생성망 모델(GAN model).

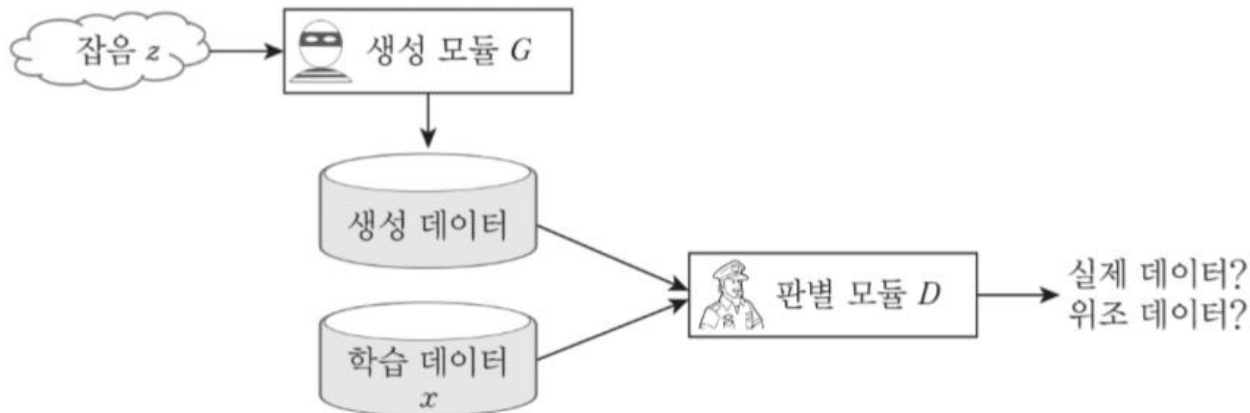
3.3.1 대립쌍 생성망의 구조

❖ 생성 모듈 G

- 데이터를 생성하는 모듈
- 위조범 역할 : 잡음 z 를 사용하여 데이터 $G(z)$ 생성

❖ 판별 모듈 D

- 생성된 데이터가 학습데이터의 특성을 갖는 평가하여 판단하는 모듈
- 경찰관 역할 : $D(x)$ 계산
 - 학습 데이터 x 에 대한 큰 값: $D(x)$ 가 큰 값
 - 생성된 데이터 $G(z)$ 에 대해서 작은 값
 - » $D(G(z))$ 의 값은 작음



3.3.2 대립쌍 생성망의 학습

❖ 생성 모듈과 판별 모듈의 동시 학습

▪ 생성 모듈 G

- 판별 모듈을 속이도록 학습
- 생성 모듈이 생성한 데이터 $G(z)$ 에 대해 판별 모듈이 큰 값을 주도록 학습
 - 잡음 $z \sim P_z(z)$ 으로 부터 생성한 데이터 $G(z)$ 에 대해 $D(G(z))$ 이 커지도록 학습

▪ 판별 모듈 D

- 생성된 데이터(가짜 데이터)를 잘 식별하도록 학습
- 학습 데이터 $x \sim P_{data}$ 에 판별 모듈의 출력값 $D(x)$ 는 크고, 생성된 데이터 $G(z)$ 에 대한 $1 - D(G(z))$ 의 값은 커지도록 학습

▪ 두 명이 하는 게임(two-player game)과 유사

▪ 최대화시킬 목적 함수

$$L(D, G) = E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

대립쌍 생성망의 학습

알고리즘 5.2 GAN 학습 알고리즘

입력 : 잡음 생성을 위한 사전 확률분포 $p_g(z)$, 학습 데이터 확률분포 $p_{data}(x)$

출력 : 학습된 GAN의 생성 모듈과 판별 모듈

1. repeat

2. for $i = 1$ to k { // 판별 모듈 학습

3. $p_g(z)$ 를 사용해 m 개의 잡음 표본으로 미니배치 구성 $\{z^{(1)}, \dots, z^{(m)}\}$

4. $p_{data}(x)$ 를 사용해 m 개의 학습 데이터로 미니배치 구성 $\{x^{(1)}, \dots, x^{(m)}\}$

5. 다음의 그래디언트 평균을 사용하여 판별 모듈의 모수 θ_d 학습

$$6. \quad \theta_d \leftarrow \theta_d + \eta \nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m [\log D(x^{(i)}) + \log (1 - D(G(z^{(i)})))]$$

7. }

8. $p_g(z)$ 를 사용하여 m 개의 잡음 표본으로 미니배치 구성 $\{z^{(1)}, \dots, z^{(m)}\}$

9. 다음의 그래디언트 평균을 사용하여 생성 모듈의 모수 θ_g 학습

$$10. \quad \theta_g \leftarrow \theta_g - \eta \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m \log (1 - D(G(z^{(i)})))$$

11. until (종료조건 만족)

대립쌍 생성망의 학습

❖ 대립쌍 생성망(GAN)의 학습과정

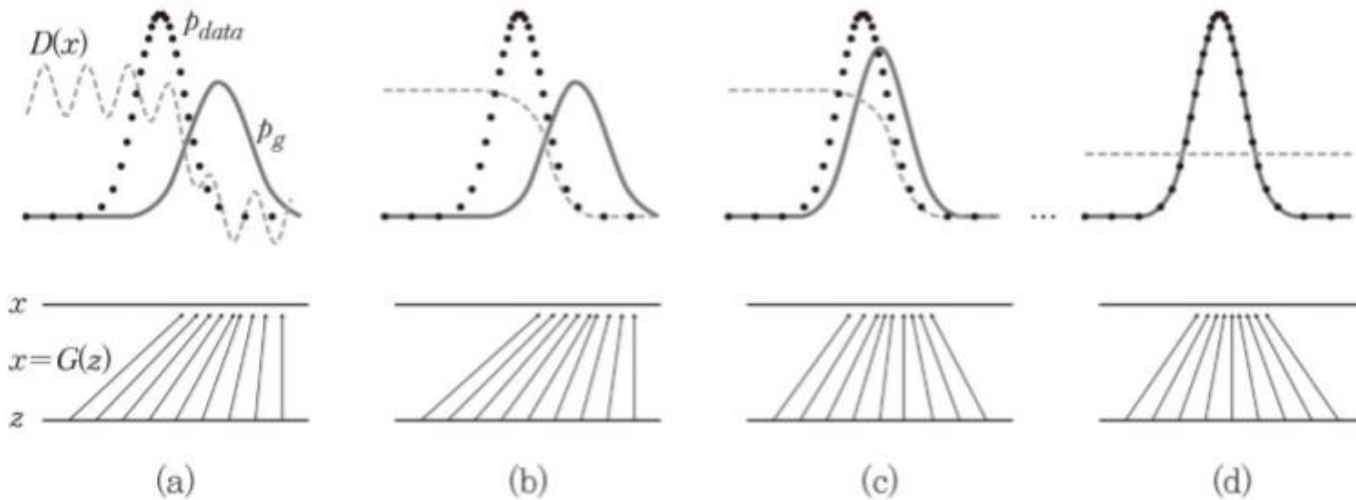


그림 5.45 대립쌍 생성망 모델의 적용 예 [출처: Goodfellow 등, 2015]

학습과정에서의 생성 모듈 $G(z)$ 에 의한 데이터 분포 p_g 와 학습 데이터 분포 p_{data} 와 판별 모듈 $D(x)$ 의 변화

P_g : 생성 모듈 $G(z)$ 에 의해서 생성되는 데이터 분포

P_{data} : 학습 데이터의 분포

$D(x)$: 판별 모듈의 출력값

대립쌍 생성망의 학습

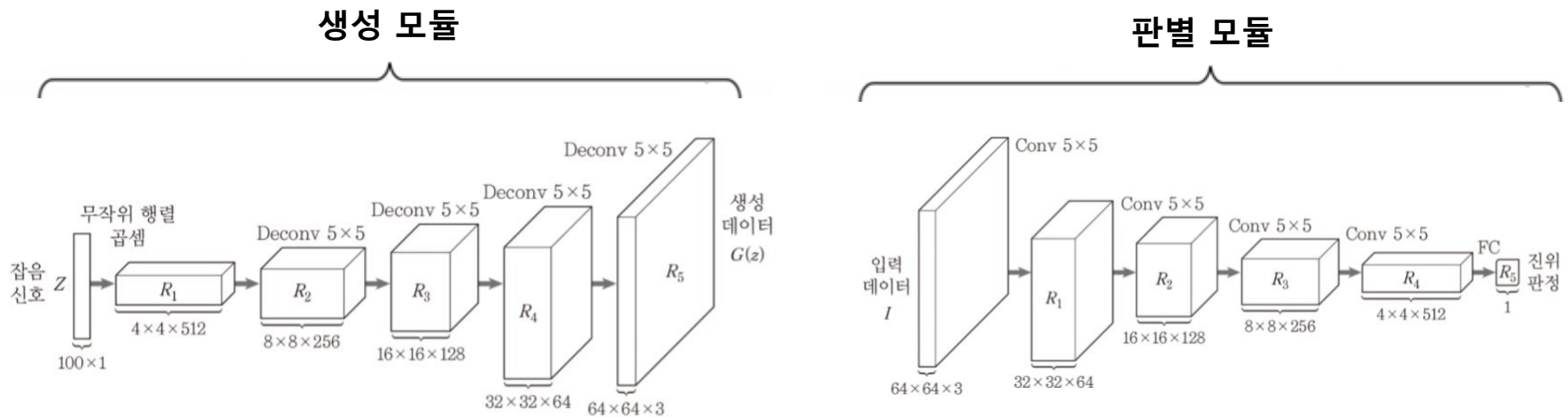
❖ 대립쌍 생성망(GAN)에 의해 생성된 이미지의 예



3.3.3 딥 컨볼루션 대립쌍 생성망 DCGAN

❖ DCGAN(deep convolutional generative adversarial networks)

- 학습 속도 및 안정성이 개선된 모델
- 생성 모듈
 - 디컨볼루션(deconvolution) 층으로 구성
- 판별 모듈
 - 생성 모듈과 비슷한 개수의 컨볼루션 층으로 구성



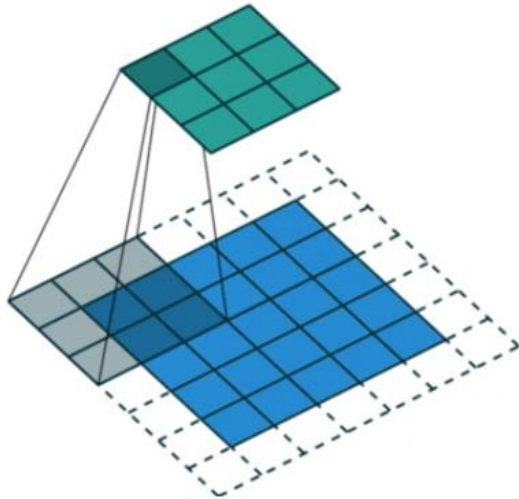
딥 컨볼루션 대립쌍 생성망 DCGAN

❖ 디컨볼루션 (deconvolution)

- 입력보다 확장된 출력 생성

컨볼루션

Output = 3x3

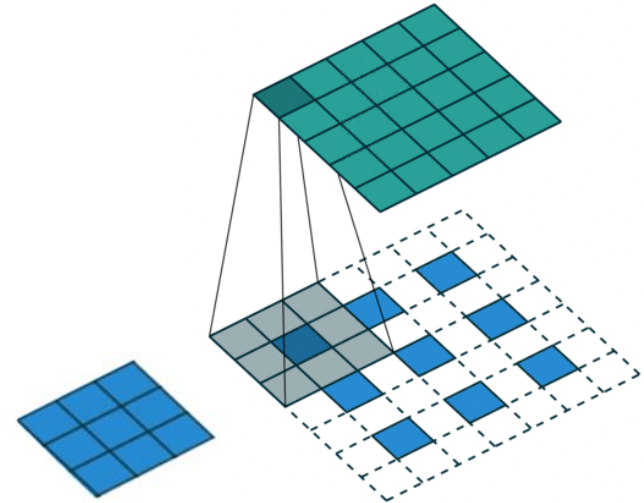


Input = 5x5
with zero-padding at
border = 6x6
(stride=2)

필터크기
3 × 3

디컨볼루션

Output = 5x5



Input = 3x3
Interlace zero-padding with
inputs = 7x7
(stride=1)

딥 컨볼루션 대립쌍 생성망 DCGAN

❖ 생성 모듈 $G(z)$

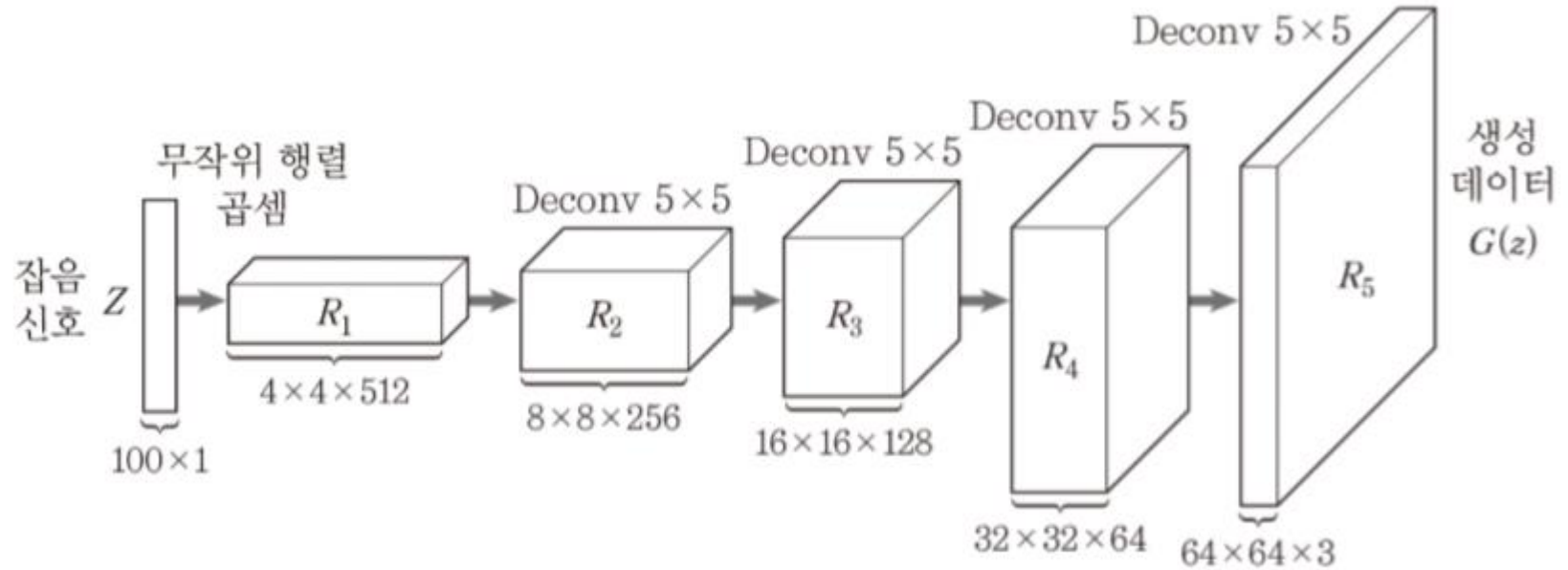


그림 5.46 DCGAN 모델의 생성 모듈

$$R_1 \leftarrow \text{ReLU}(\text{BN}(z M_R))$$

$$R_i \leftarrow \text{ReLU}(\text{BN}(\text{Deconv}(R_{i-1}))) \quad i = 2, 3, 4$$

$$R_5 \leftarrow \tanh(\text{Deconv}(R_4))$$

딥 컨볼루션 대립쌍 생성망 DCGAN

❖ 판별 모듈 $D(x)$

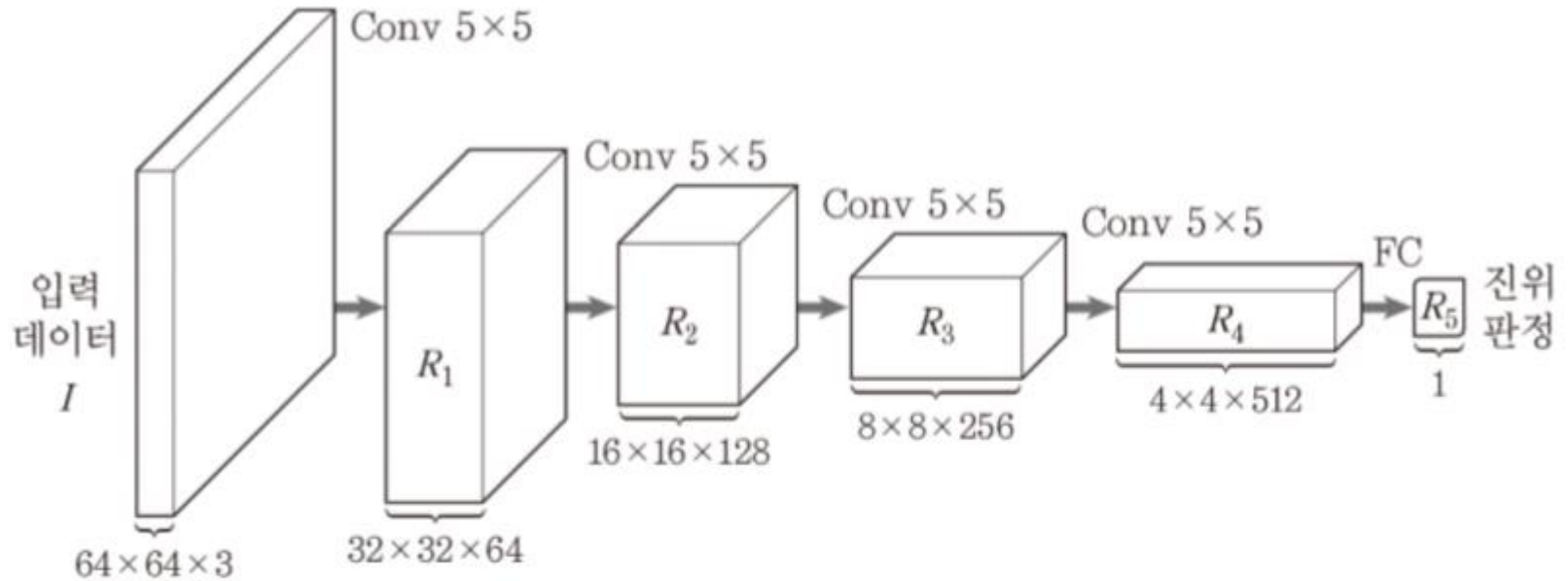
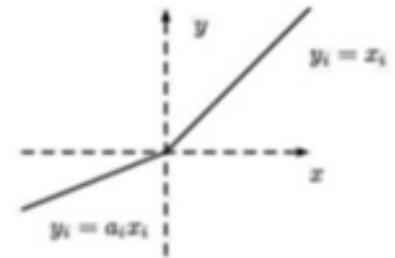


그림 5.48 DCGAN 모델의 판별 모듈

$$R_1 \leftarrow \text{lReLU}(\text{Conv}(I)) \quad \text{누수 ReLU 사용}$$

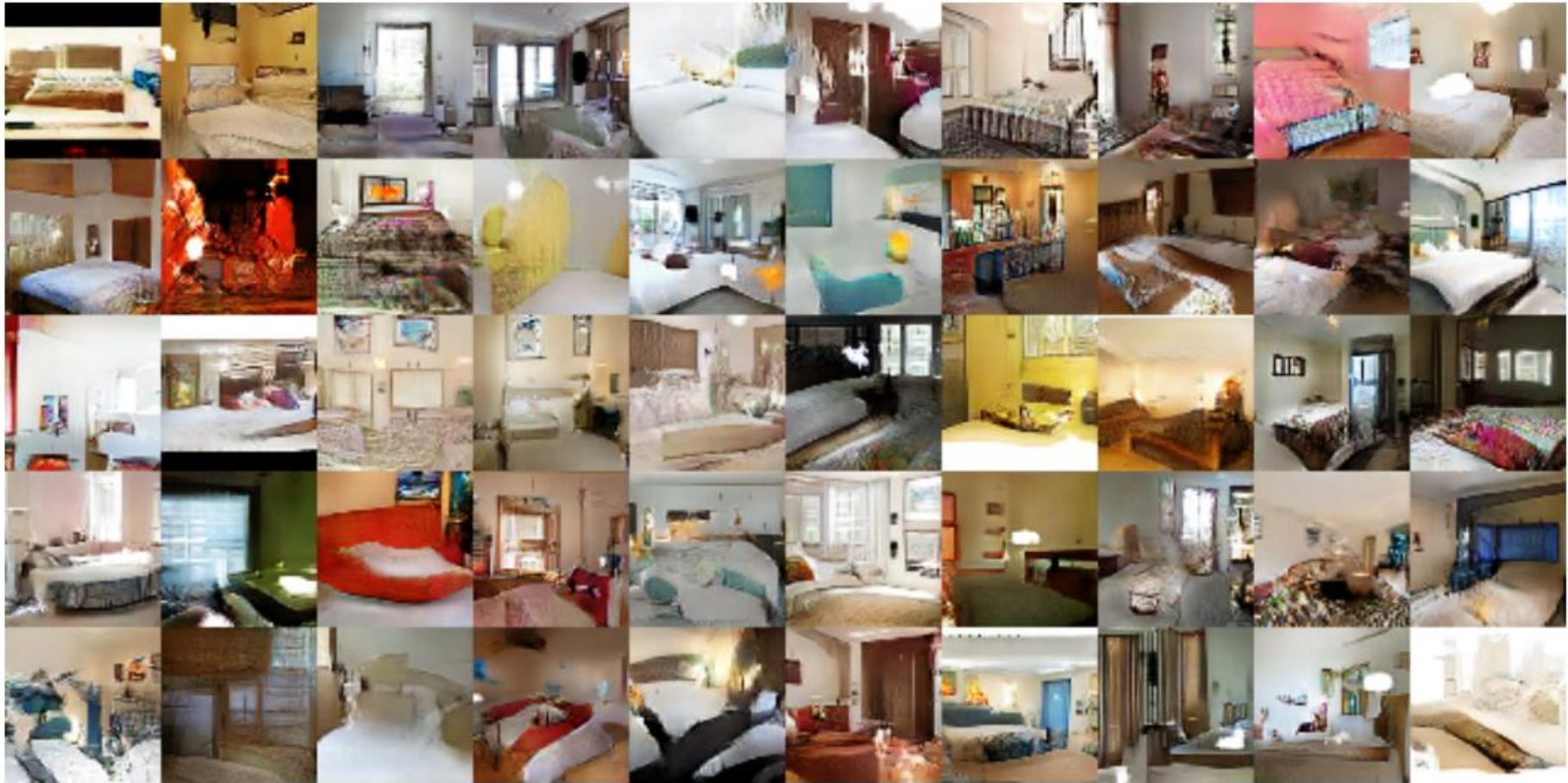
$$R_i \leftarrow \text{lReLU}(\text{BN}(\text{Conv}(R_{i-1}))) \quad i = 2, 3, 4$$

$$R_5 \leftarrow \sigma(\text{FC}(R_4))$$



딥 컨볼루션 대립쌍 생성망 DCGAN

❖ DCGAN으로 생성한 이미지의 예



딥 컨볼루션 대립쌍 생성망 DCGAN

❖ 잡음 벡터의 연산

- 잡음(noise)을 이미지를 코딩하는 은닉변수(latent variable)로 해석 가능
- 유사한 이미지를 생성하는 잡음 벡터들을 선택
- 잡음 벡터들의 연산을 통한 특정 특징의 추가 및 삭제 가능
 - '미소짓는 여성'들의 잡음 벡터 z_{sw} - '무표정한 여성'들의 잡음 벡터 z_{nw} + '무표정한 남성'의 잡음 벡터 z_{nm} → '미소짓는 남성'의 잡음 벡터 z_{sm}

$$z_{sm} \leftarrow z_{sw} - z_{nw} + z_{nm}$$



3.3.4 양방향 대립쌍 생성망 BiGAN

❖ 양방향 대립쌍 생성망 (Bidirectional GAN, BiGAN)

- 데이터의 역사상(inverse mapping)을 함께 고려한 대립쌍 생성성
 - 판별 모듈에서 데이터 공간의 벡터와 은닉공간의 벡터를 함께 사용

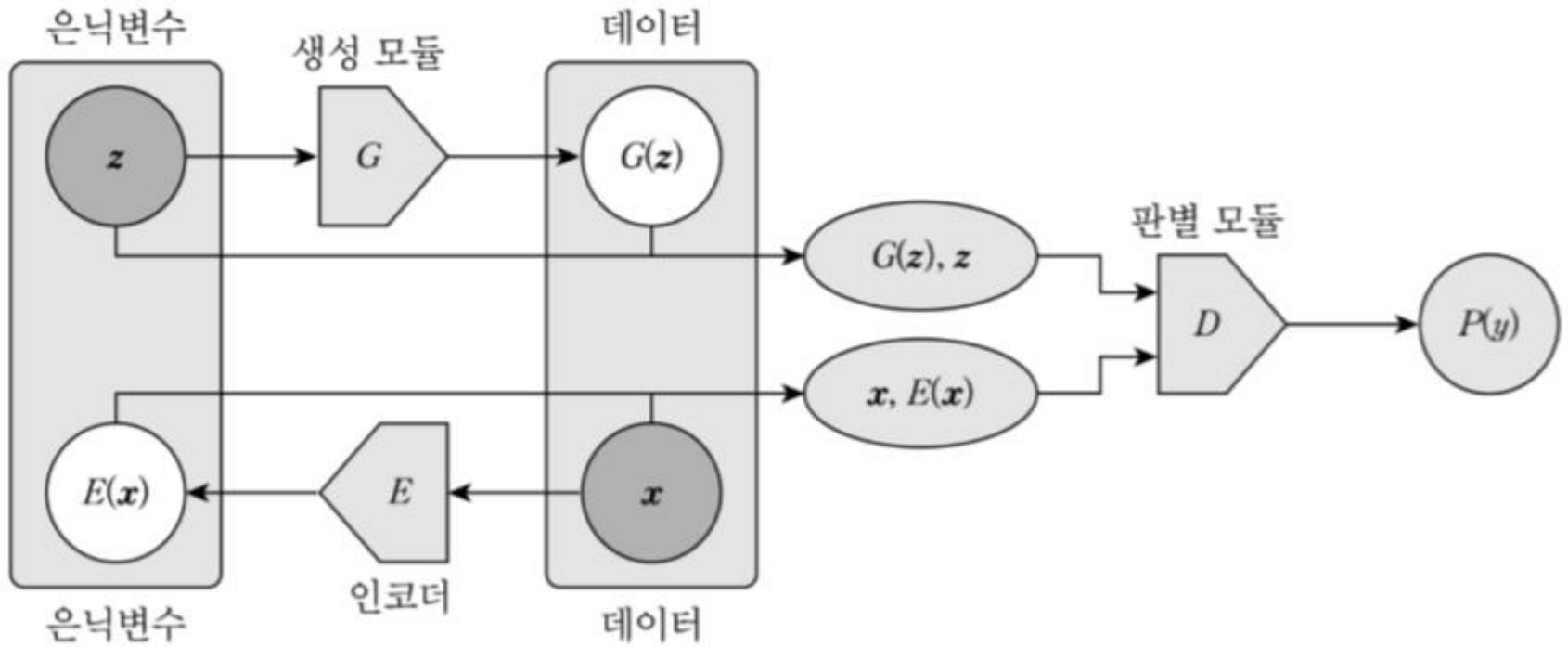
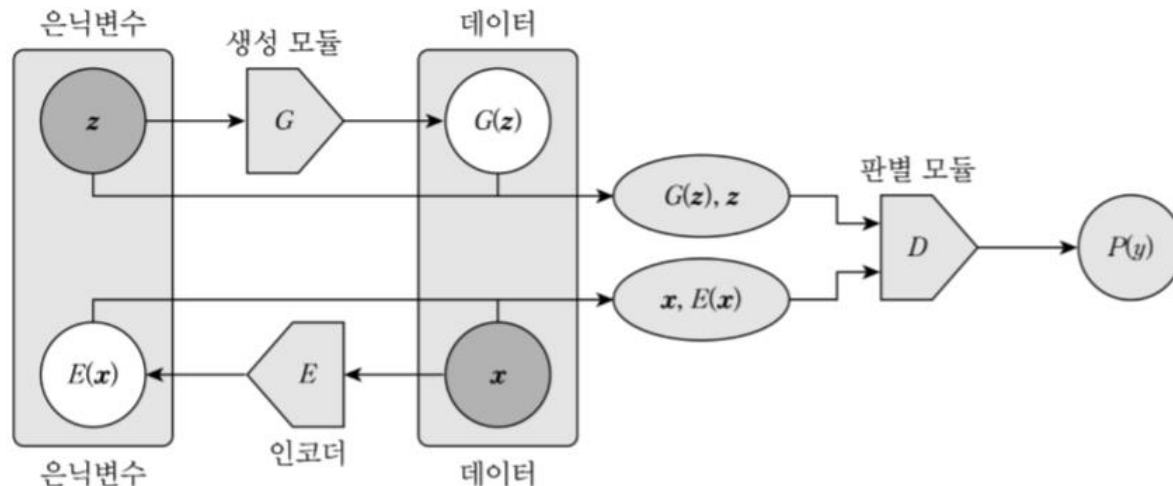


그림 5.50 BiGAN 모델의 구성

양방향 대립쌍 생성망 BiGAN

❖ BiGAN의 학습에서 사용하는 목적함수

- 인코더
 - 데이터 x 를 은닉 공간의 $E(x)$ 로 사상(mapping)
- 판별 모듈
 - 은닉공간 벡터와 데이터를 함께 입력으로 받음
 - 진위(眞僞)를 판정



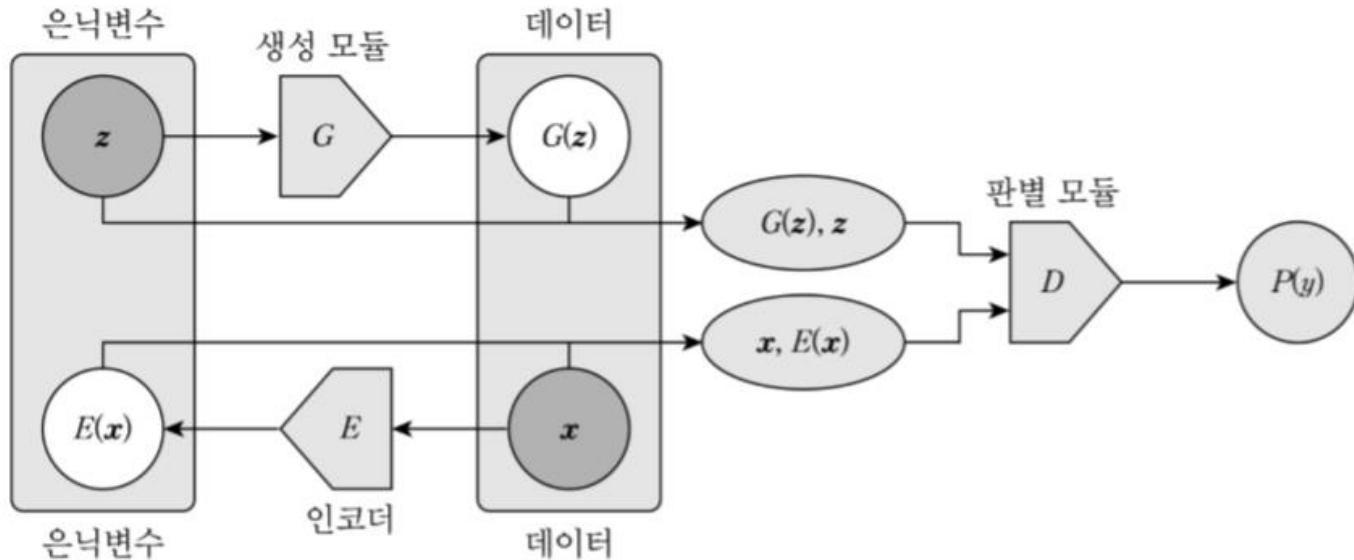
▪ 목적함수

$$L(D, E, G) = E_{x \sim p_{data}(x)} [\log D(x, E(x))] + E_{z \sim p_z(z)} [\log(1 - D(G(x), z))]$$

양방향 대립쌍 생성망 BiGAN

❖ BiGAN의 학습

- 목적 함수의 최대화를 위해 판별 모듈 학습
- 목적 함수의 최대화를 위해 인코더 학습
- 목적 함수의 최소화를 위해 생성 모듈 학습



$$L(D, E, G) = E_{x \sim p_{data}(x)} [\log D(x, E(x))] + E_{z \sim p_z(z)} [\log(1 - D(G(x), z))]$$

양방향 대립쌍 생성망 BiGAN

❖ BiGAN을 이용한 인코딩과 데이터 생성의 예

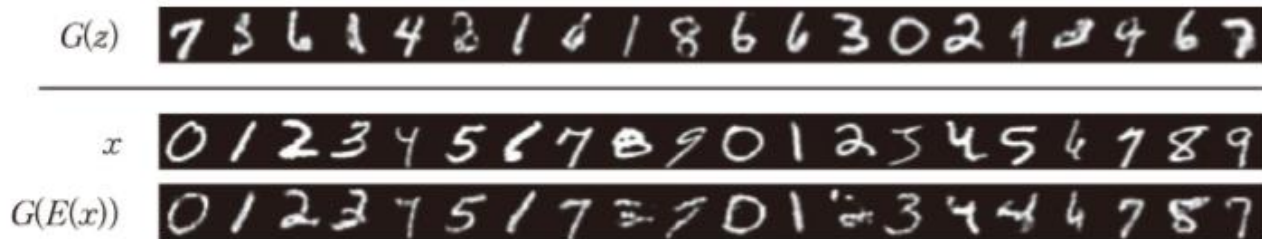
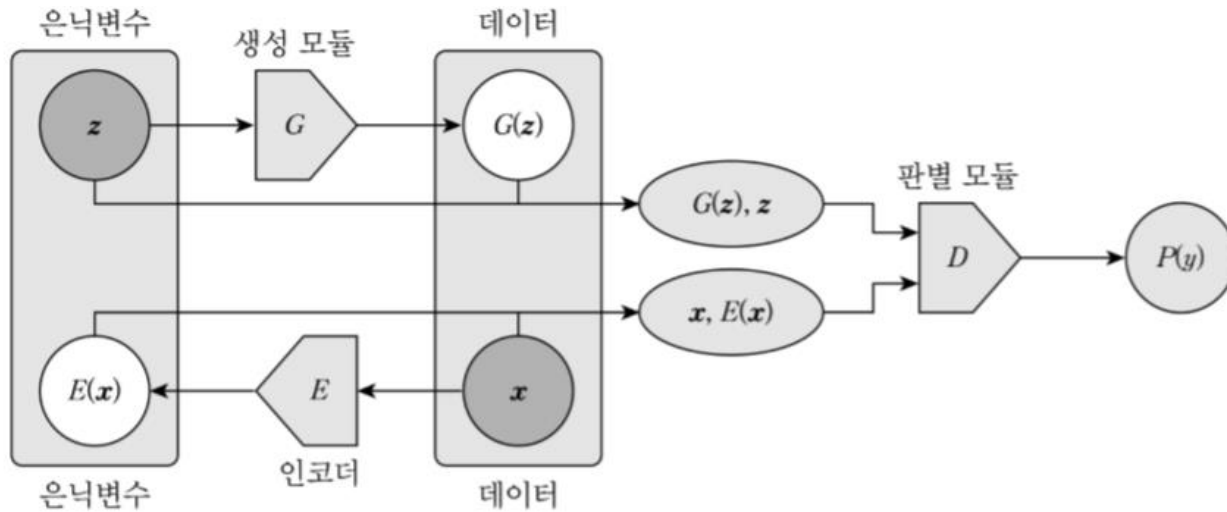


그림 5.51 생성 모듈의 출력 예 $G(z)$, 실제 데이터 x 와 인코더로 생성된 은닉 공간의 값 $E(x)$ 로부터 생성된 데이터 $G(E(x))$ 의 비교 [출처 Donahue 등, 2016]