A Fast Learning Algorithm for Deep Belief Nets

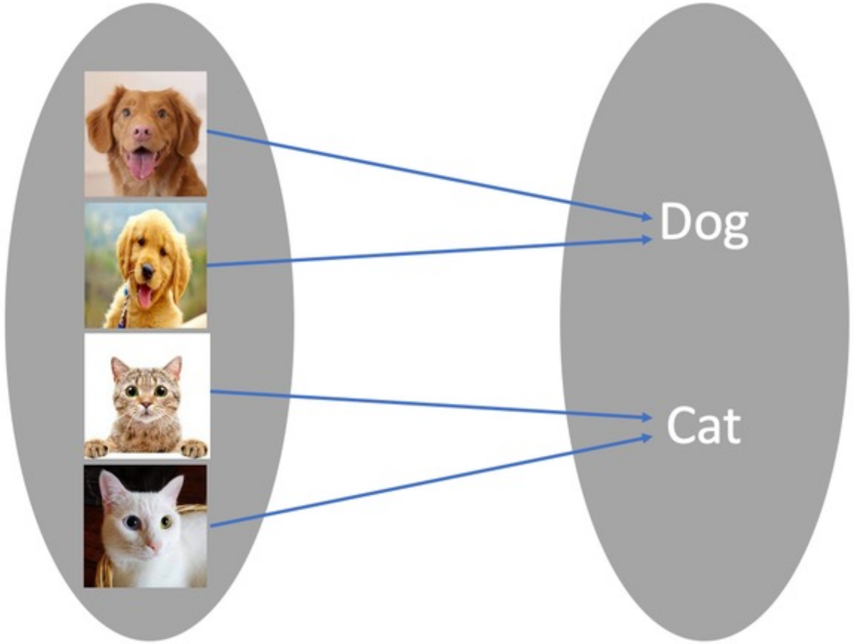
Generative vs Discriminative

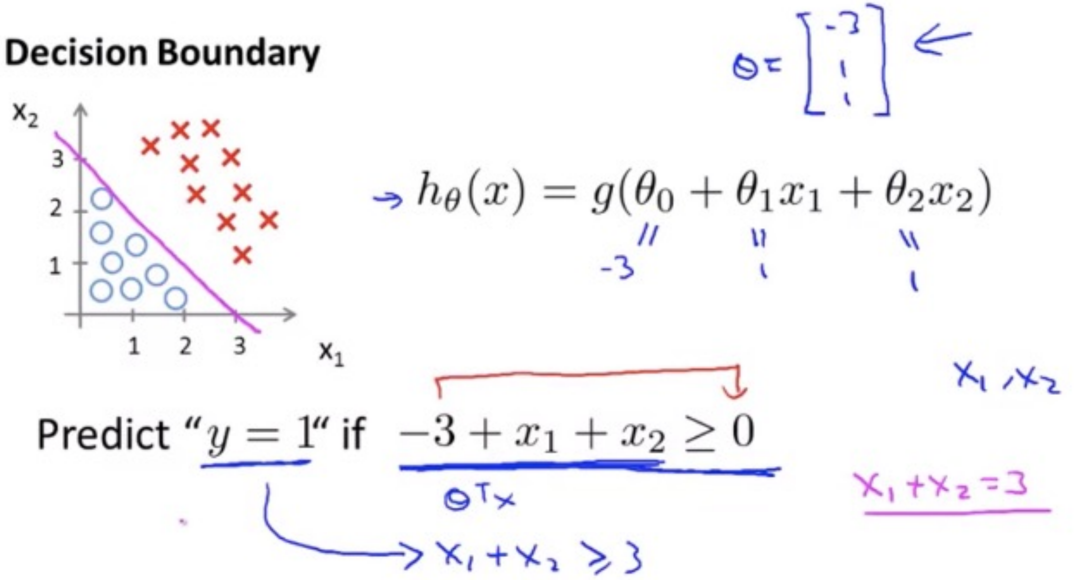
ML의 기본적인 목표 - Discriminative

Input X가 주어졌을 때 그 결과 Y일 확률을 알아 내는 것

P(Y|X)를 직접적으로 활용

X라는 값들이 충분히 의미있는 데이터일때 X를 기반으로 Y가 0 or 1일지 확률적으로 예측





Generative vs Discriminative

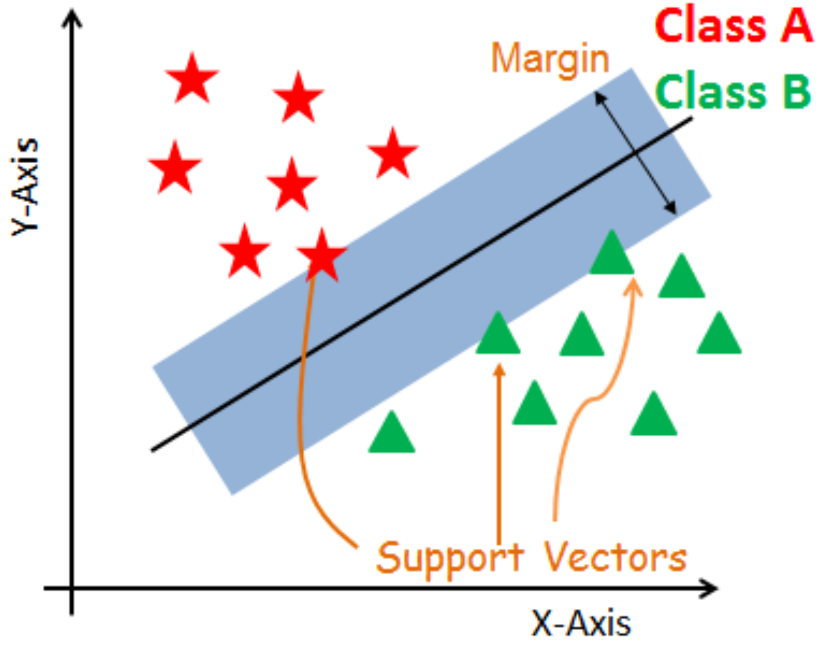
ML의 기본적인 목표 - Discriminative

장점 : 데이터가 충분할 경우 성능이 좋다

단점 : 데이터의 분포, 즉 데이터가 실제 어떤 모습인지는 이해할 수 없다. 오로지 데이터를 구분하는데 목표를 둔다.

결국 Decision Boundary를 학습하는 것이 목표이다!

EX) linear, logistic regression, svm …

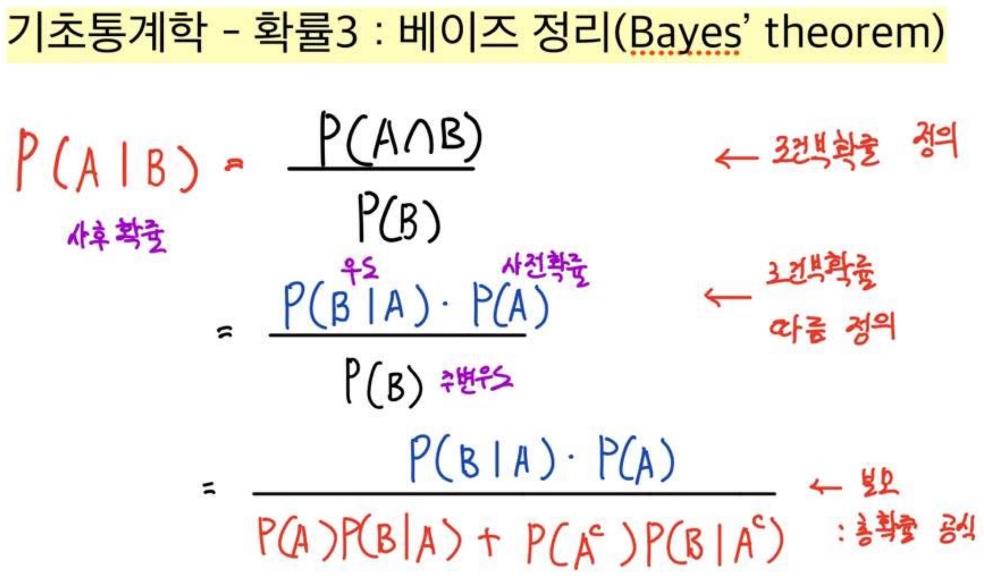


Generative vs Discriminative

Supervised Generative model

Input X가 주어졌을 때 그 결과 Y일 확률을 알아 내는 것에 더해서 class에 맞는 데이터를 생성해 낼 수 있는 방식

P(Y|X) = P(X|Y) * P(Y) / P(X) → P(X,Y) = P(X|Y) * P(Y)



출처: https://datalabbit.tistory.com/21

Generative vs Discriminative

Supervised Generative model

Input X가 주어졌을 때 그 결과 Y일 확률을 알아 내는 것에 더해서 class에 맞는 데이터를 생성해 낼 수 있는 방식

P(Y|X) = P(X|Y) * P(Y) / P(X) → P(X,Y) = P(X|Y) * P(Y)

P(X|Y),P(Y)를 통해 데이터 분포를 학습하고 간접적으로 P(Y|X)를 구함 MLE, MAP …

장점: 데이터 셋이 적어도 쓸만 하다, 데이터 자체의 특성을 파악하기에 좋다, 데이터를 생성할 수 있다.

단점: 많은 가정이 필요하다 ex)Y의 분포에 대한 가정 등등

Ex) 나이브 베이즈 분류기, LDA

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Generative Model

Naïve Bayes 분류기

P(정상 메일 | 입력 텍스트) = 입력 텍스트가 있을 때 정상 메일일 확률
P(스팸 메일 | 입력 텍스트) = 입력 텍스트가 있을 때 스팸 메일일 확률

P(정상 메일 | 입력 텍스트) = (P(입력 텍스트 | 정상 메일) × P(정상 메일)) / P(입력 텍스트)
P(스팸 메일 | 입력 텍스트) = (P(입력 텍스트 | 스팸 메일) × P(스팸 메일)) / P(입력 텍스트)

P(Y|X) = P(X|Y) * P(Y) / P(X)

P(정상 메일 | 입력 텍스트) = P(w_1 | 정상 메일) × P(w_2 | 정상 메일) × P(w_3 | 정상 메일) × P(정상 메일)
P(스팸 메일 | 입력 텍스트) = P(w_1 | 스팸 메일) × P(w_2 | 스팸 메일) × P(w_3 | 스팸 메일) × P(스팸 메일)

← Feature 즉, 단어들이 모두 독립적으로 추출된다고 가정

P(Y|X) = P(X|Y) * P(Y) / P(X)
→ P(X)는 비교하는데 사용되지 않는다!
→ P(Y|X) = P(X|Y) * P(Y)

출처: https://wikidocs.net/22892

Generative Model

Naïve Bayes 분류기

	메일로부터 토큰화 및 정제 된 단어들	분류
1	me free lottery	스팸 메일
2	free get free you	스팸 메일
3	you free scholarship	정상 메일
4	free to contact me	정상 메일
5	you won award	정상 메일
6	you ticket lottery	스팸 메일

Input : you free lottery

P(정상 메일 | 입력 텍스트) = P(you | 정상 메일) × P(free | 정상 메일) × P(lottery | 정상 메일) × P(정상 메일)

P(스팸 메일 | 입력 텍스트) = P(you | 스팸 메일) × P(free | 스팸 메일) × P(lottery | 스팸 메일) × P(스팸 메일)

P(정상 메일) = P(스팸 메일) = 총 메일 6개 중 3개 = 0.5

P(정상 메일 | 입력 텍스트) = 2/10 × 2/10 × 0/10 = 0

P(스팸 메일 | 입력 텍스트) = 2/10 × 3/10 × 2/10 = 0.012

아무튼.. P(Y|X)를 알아내기 위해 P(X|Y) * P(Y) 를 이용하여 간접적으로 판별 → P(X|Y) * P(Y) = P(X,Y)

출처: https://wikidocs.net/22892

Generative vs Discriminative

Unsupervised Generative model - GMM

$$p(x) = \sum_{i=1}^{k} \phi_i \mathcal{N}(x|\mu_i, \Sigma_i)$$

GMM 수식

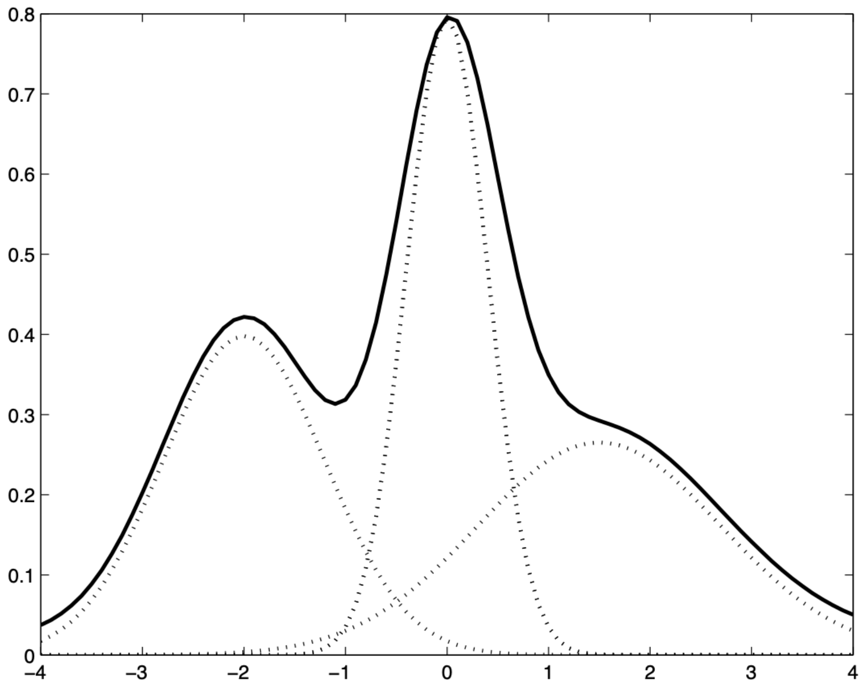
- $p(x)$: 데이터 포인트 x의 확률
- ϕi : i 번째 Gaussian 분포의 합의 weight (weight의 합은 1)
- $N(x|\mu i,\Sigma i)$: i번째 Gaussian 분포의 확률 밀도 함수
- μi : i번째 Gaussian 분포의 평균
- Σi : i번째 Gaussian 분포의 공분산 행렬

- GMM Clustering 알고리즘은 결국 GMM 수식에서 ϕi (각 Gaussian 분포의 weight), μi (각 Gaussian 분포의 평균), Σi (각 Gaussian 분포의 공분산 행렬)을 찾는 과정이다.

출처: https://devhwi.tistory.com/8

Generative vs Discriminative

Unsupervised Generative model - GMM

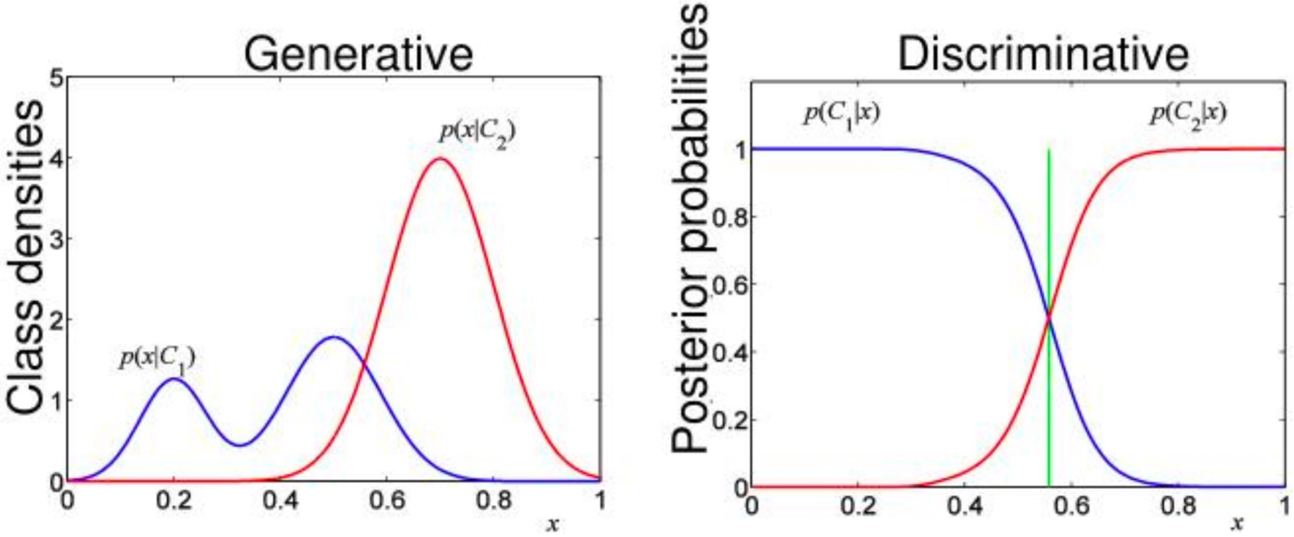


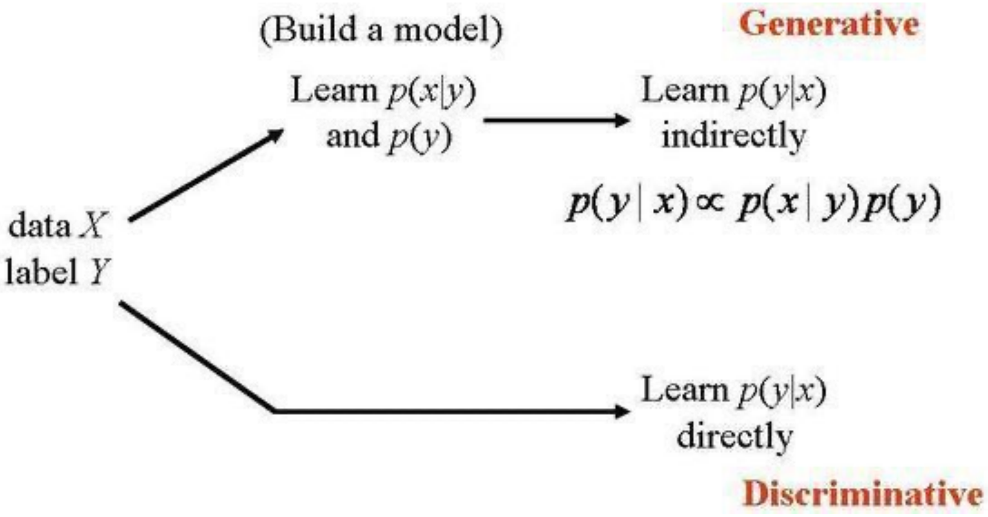
파라미터를 찾기 위해 EM 알고리즘을 이용.

아무튼 결과적으로 P(X) 즉, 데이터의 분포를 여러 개의 gaussian 분포가 합쳐져 있다고 생각하고 모델링

출처: https://ratsgo.github.io/speechbook/docs/am/gmm

Generative vs Discriminative

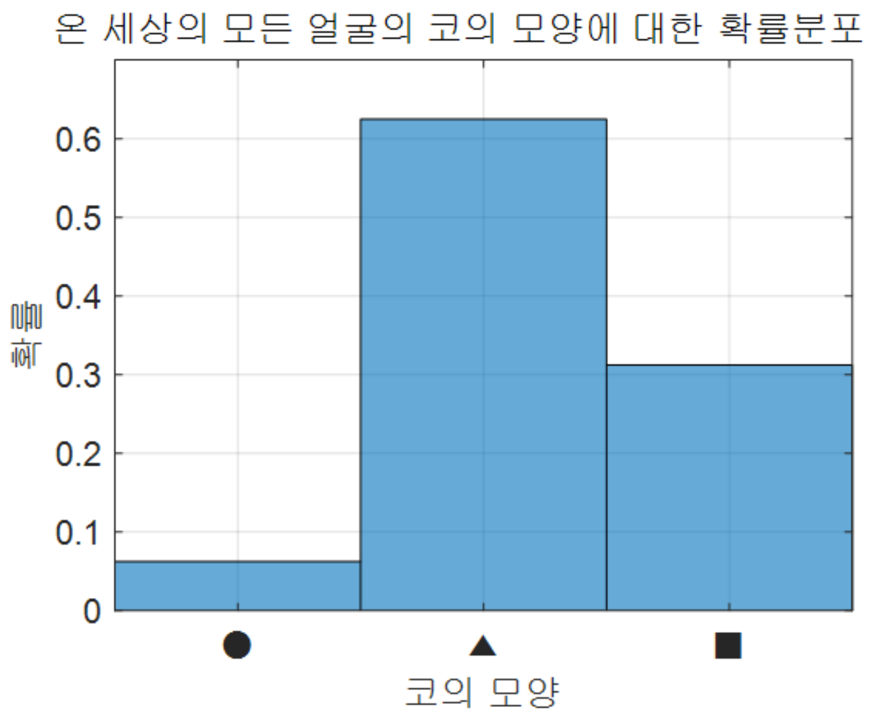




Q & A

Restricted Boltzmann Machine(RBM)

RBM은 생성모델이다!



사람의 얼굴 중에서 코만 그려주는 모델이 있다고 할 때, 모델이 학습한 코의 분포가 왼쪽과 같다면, 기계는 세모 모양의 코를 그려줄 가능성이 높다.

→ 코 뿐만 아니라 얼굴 전체에 대한 분포를 잘 학습했다면? 얼굴을 일리있게 생성 할 수 있게 된다.

출처: https://angeloyeo.github.io/2020/10/02/RBM.html

Restricted Boltzmann Machine(RBM)

RBM은 생성모델이다!

노란색이 visible unit, 초록색이 hidden unit

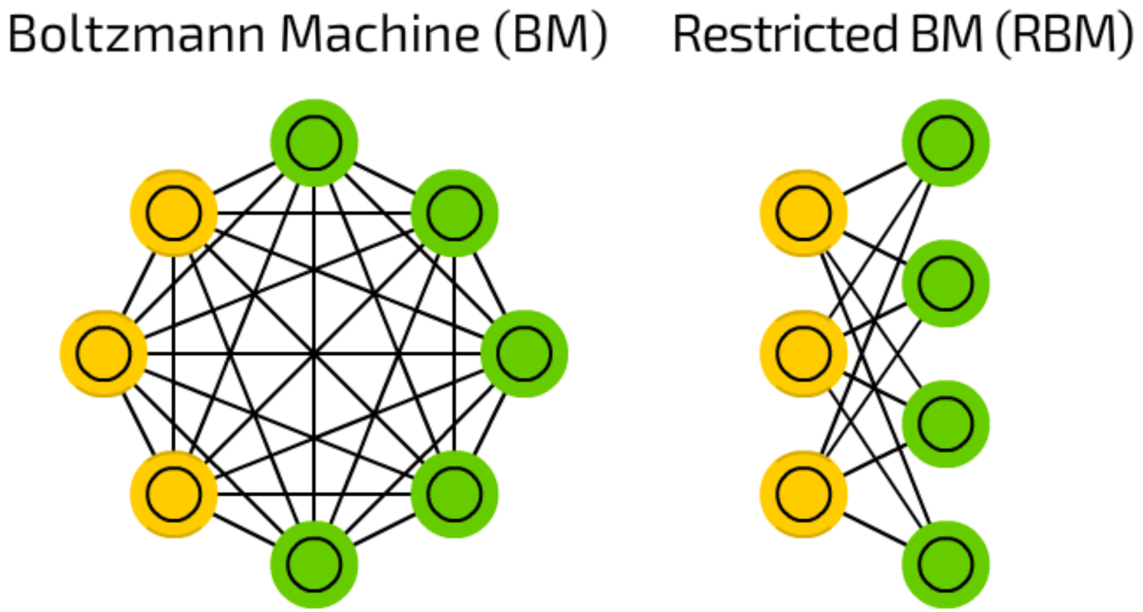


그림 3. Boltzmann Machine과 Restricted Boltzmann Machine

출처: https://angeloyeo.github.io/2020/10/02/RBM.html

Restricted Boltzmann Machine(RBM)

RBM은 생성모델이다!

RBM은 우리가 직접적으로 볼 수 없는 요소들 까지도 잘 학습 시킨다면 확률 분포를 좀더 명확하게 학습시킬 수 있지 않을까? 라는 가정으로 출발 → Hidden layer가 존재

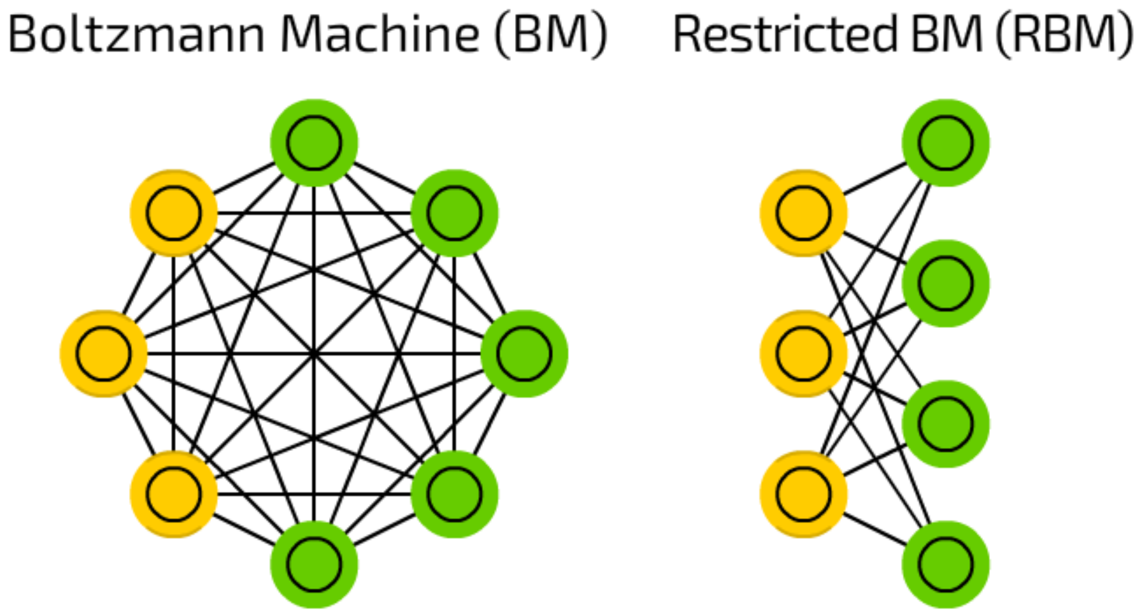
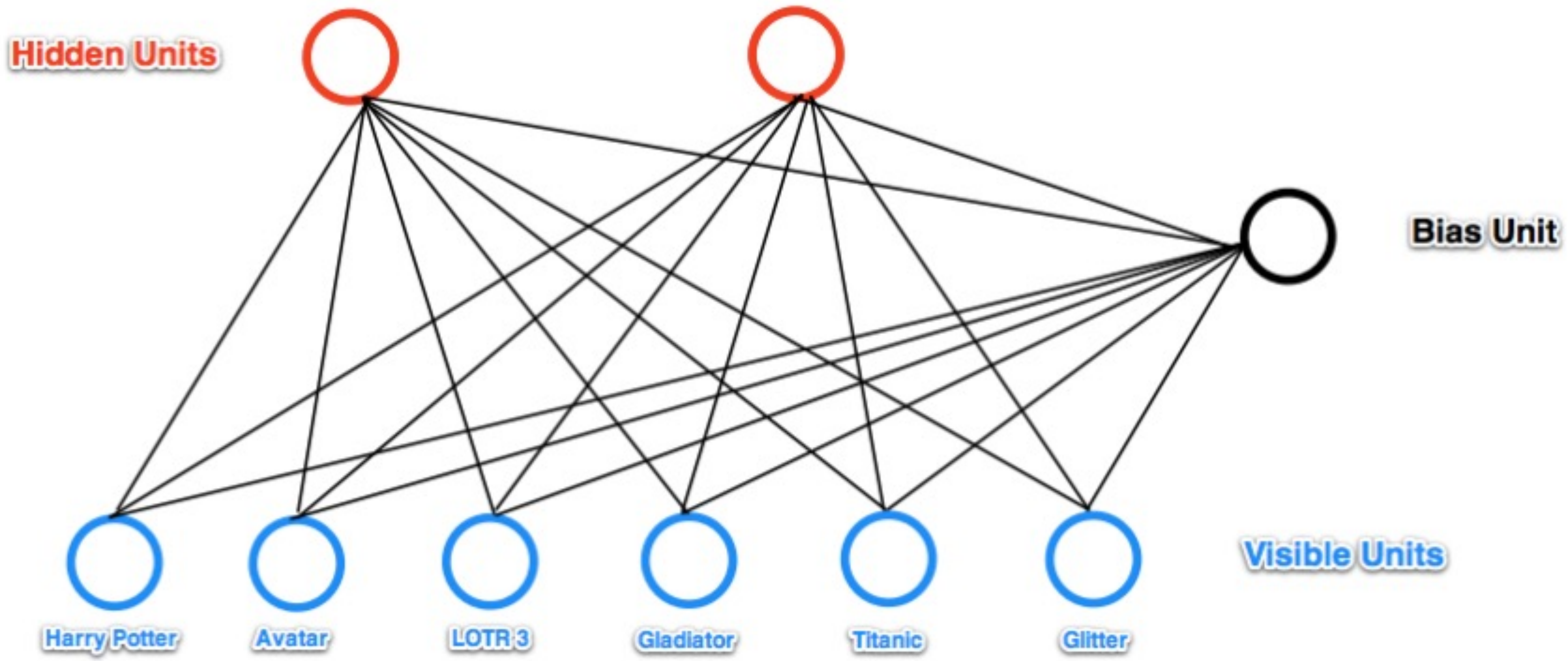


그림 3. Boltzmann Machine과 Restricted Boltzmann Machine

출처: https://angeloyeo.github.io/2020/10/02/RBM.html

Restricted Boltzmann Machine(RBM)

RBM은 생성모델이다!



출처: https://angeloyeo.github.io/2020/10/02/RBM.html

Restricted Boltzmann Machine(RBM)

RBM은 생성모델이다!

BM은 계산을 할때 Joint Distribution을 계산해야함

즉, P(X,H) = P(x1,x2,x3,h1,h2,h3…)

이런 계산 복잡도를 낮추기 위해 Visible과 Hidden끼리의 연결을 없애 P(H|X)나 P(X|H)로 계산 할 수 있게 만든 것이 RBM → Feed Forward Network처럼 학습

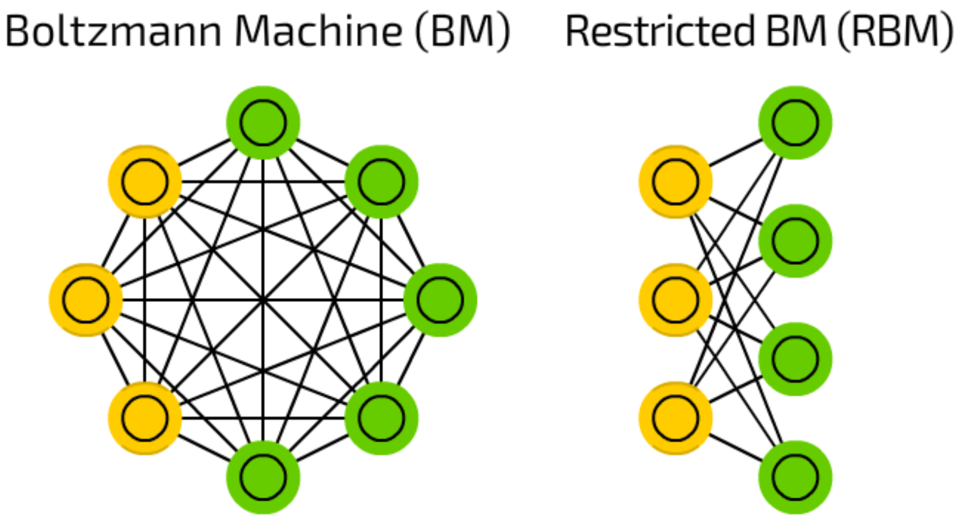
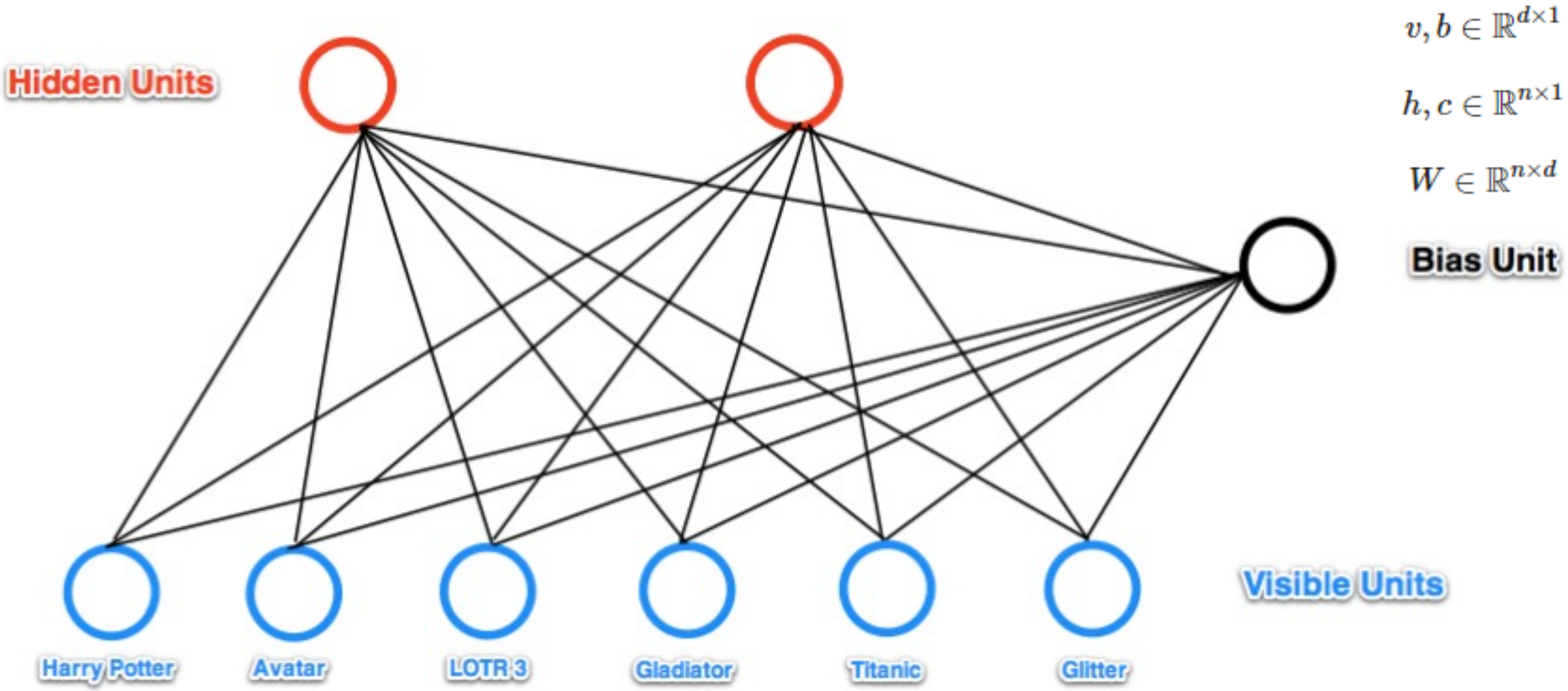


그림 3. Boltzmann Machine과 Restricted Boltzmann Machine

출처: https://angeloyeo.github.io/2020/10/02/RBM.html

Restricted Boltzmann Machine(RBM)

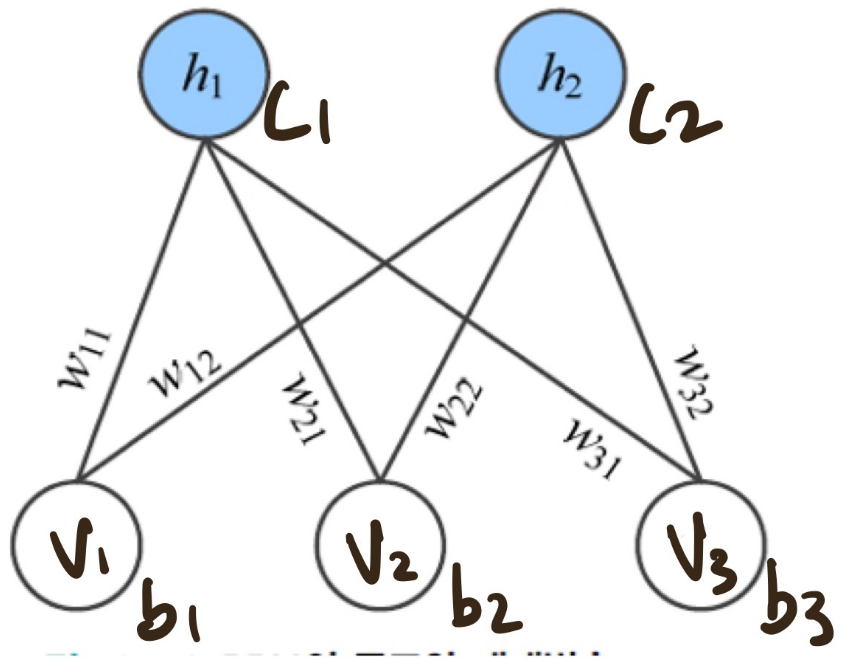
RBM은 생성모델이다! – 학습 예시



출처: https://angeloyeo.github.io/2020/10/02/RBM.html

Restricted Boltzmann Machine(RBM)

RBM은 생성모델이다! – 구성 파라미터 // h와 v는 0 or 1의 값을 갖는다고 가정



$v, b \in \mathbb{R}^{d\times 1}$

$h, c \in \mathbb{R}^{n\times 1}$

$W \in \mathbb{R}^{n\times d}$

$$\mathbf{W} = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{pmatrix}$$

출처: 오일석 교수님 ML

Restricted Boltzmann Machine(RBM)

학습이 잘 됐는지 어떻게 알까? – Energy based model // h와 v는 0 or 1의 값을 갖는다고 가정

숫자에 대한 input P(v)에 대한 확률은 높이는 방향으로 학습을 진행

→ Input에 대한 energy가 낮아지도록 학습을 진행

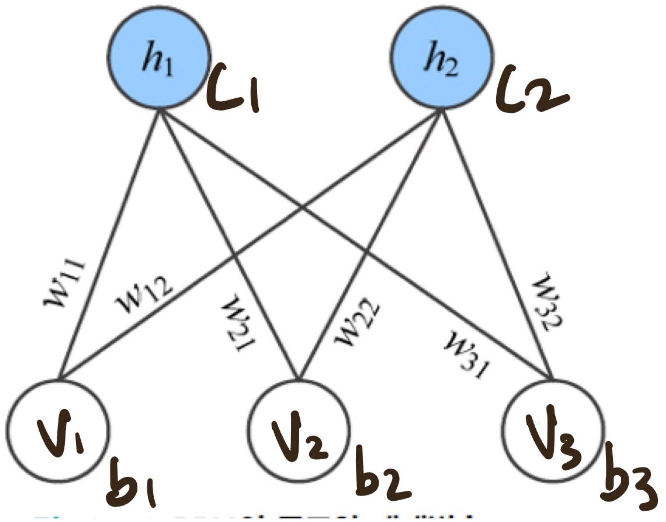
출처: https://angeloyeo.github.io/2020/10/02/RBM.html

Restricted Boltzmann Machine(RBM)

학습이 잘 됐는지 어떻게 알까? – Energy based model // h와 v는 0 or 1의 값을 갖는다고 가정

$$E(v,h) = -b^T v - c^T h - h^T W v$$

← Energy, 에너지 값이 낮을 수록 안정적인 상태이다
즉, energy가 낮을 수록 input과 재구성된 data가 유사할 수 있도록 학습이 진행



$$\mathbf{W} = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{pmatrix}$$

출처: https://angeloyeo.github.io/2020/10/02/RBM.html

Restricted Boltzmann Machine(RBM)

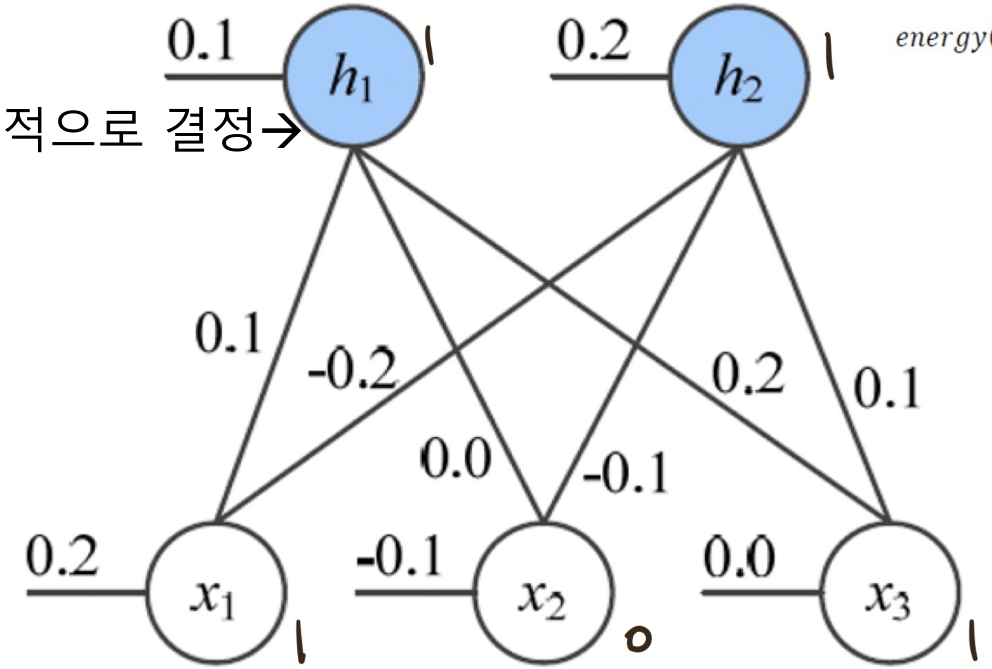
학습이 잘 됐는지 어떻게 알까? – Energy based model // h와 v는 0 or 1의 값을 갖는다고 가정

$$E(v,h) = -b^T v - c^T h - h^T W v$$

← Energy, 에너지 값이 낮을 수록 안정적인 상태이다
즉, energy가 낮을 수록 input과 재구성된 data가 유사할 수 있도록 학습이 진행

$$\begin{aligned} energy(\mathbf{x} = (1,0,1)^{\mathrm{T}}, \mathbf{h} = (1,1)^{\mathrm{T}}) &= -(1 \times 0.2 + 0 \times (-0.1) + 1 \times 0) - (1 \times 0.1 + 1 \times 0.2) - (1 \times 1 \times 0.1 + 1 \times 1 \\ &\quad \times (-0.2) + 0 \times 1 \times 0.0 + 0 \times 1 \times (-0.1) + 1 \times 1 \times 0.2 + 1 \times 1 \times 0.1) = -0.7 \end{aligned}$$

확률적으로 결정→



출처: https://angeloyeo.github.io/2020/10/02/RBM.html

Restricted Boltzmann Machine(RBM)

학습이 잘 되는지는 어떻게 알까? – Energy based model // h와 v는 0 or 1의 값을 갖는다고 가정

$$E(v,h) = -b^T v - c^T h - h^T W v$$

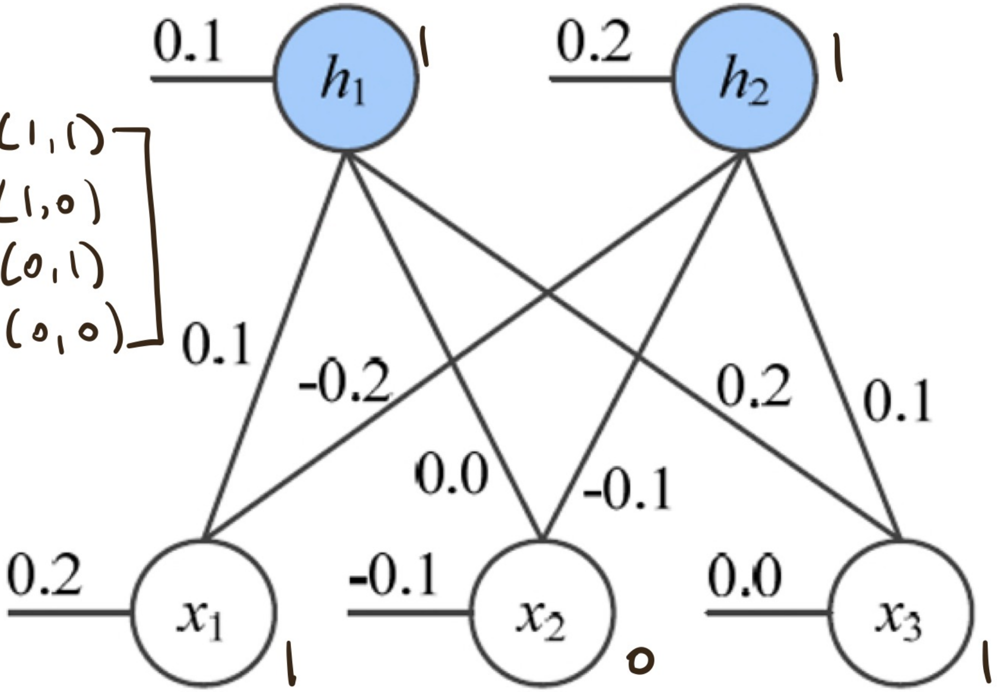
← But 얼마나 낮아야 잘 학습이 된 건지 알 수 없다.
확률로 바꿈

Energy를 통해 P(v)확률을 구하는법

$$p(v) = \sum_h p(v,h) = \boxed{\sum_h} \frac{\boxed{\exp(-E(v,h))}}{Z}$$

$$\text{where } Z = \sum_v \sum_h \exp(-E(v,h))$$

$h = \begin{bmatrix} (1,1) \\ (1,0) \\ (0,1) \\ (0,0) \end{bmatrix}$



출처: https://angeloyeo.github.io/2020/10/02/RBM.html

Restricted Boltzmann Machine(RBM)

학습이 잘 되는지는 어떻게 알까? – Energy based model // h와 v는 0 or 1의 값을 갖는다고 가정

$$E(v,h) = -b^T v - c^T h - h^T W v$$

← But 얼마나 낮아야 잘 학습이 된 건지 알 수 없다.
확률로 바꿈

Energy를 통해 P(v)확률을 구하는법

$$p(v) = \sum_h p(v,h) = \sum_h \frac{\exp(-E(v,h))}{\boxed{Z}}$$

$$\text{where } \boxed{Z = \sum_v \sum_h \exp(-E(v,h))}$$

$$Z = E\begin{pmatrix} v=(0,0,0) & h=(0,0) \\ (0,0,1) & (0,1) \\ \vdots & (1,0) \\ (1,1,1) & (1,1) \end{pmatrix}$$

Z는 모든 상태에서의 Energy의 합

출처: https://angeloyeo.github.io/2020/10/02/RBM.html

Restricted Boltzmann Machine(RBM)

학습이 잘 되는지는 어떻게 알까? – Energy based model // h와 v는 0 or 1의 값을 갖는다고 가정

$$E(v,h) = -b^T v - c^T h - h^T W v$$

← But 얼마나 낮아야 잘 학습이 된 건지 알 수 없다.
확률로 바꿈

Energy를 통해 P(v)확률을 구하는법

$$p(v) = \sum_h p(v,h) = \sum_h \frac{\exp(-E(v,h))}{Z}$$

결국 모든 Energy 중에 현재 input에 대한 energy가 확률이 된다!

$$\text{where } Z = \sum_v \sum_h \exp(-E(v,h))$$

출처: https://angeloyeo.github.io/2020/10/02/RBM.html

Restricted Boltzmann Machine(RBM)

수식 전개

$$p(v) = \sum_h p(v,h) = \sum_h \frac{\exp(-E(v,h))}{Z}$$

$$\text{where } Z = \sum_v \sum_h \exp(-E(v,h))$$

$$e^{-\log \sum_h e^{-E(v,h)}} = -\sum_h e^{-E(v,h)}$$

$$\text{식 (6)} \Rightarrow p(v) = \frac{\exp(-F(v))}{Z'}$$

$$\text{where } F(v) = -\log \sum_h \exp(-E(v,h))$$

$$\text{and } Z' = \sum_v \exp(-F(v))$$

F(v)를 Free energy라고 한다.

출처: https://angeloyeo.github.io/2020/10/02/RBM.html

Restricted Boltzmann Machine(RBM)

수식 전개

$$F(v) = -\log \sum_h \exp(-E(v,h)) \Rightarrow -\log \sum_h \exp(-(-b'v - c'h - h'Wv))$$

$$= -\log \sum_h \exp(b'v + c'h + h'Wv)$$

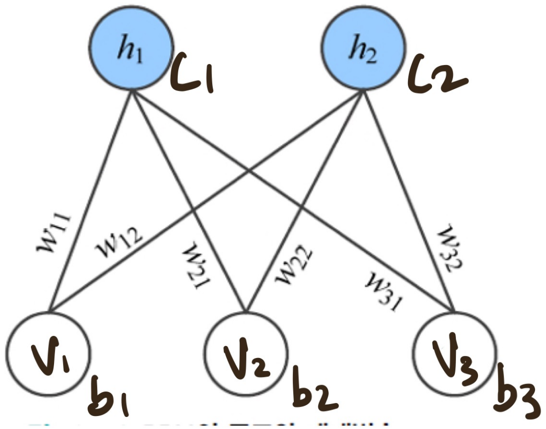
$$= -\log \sum_h \exp(b'v)\exp(c'h + h'Wv)$$

$$\Rightarrow -b'v - \log \sum_h \exp(c'h + h'Wv)$$

여기서 $h \in 0, 1$ 이므로,

$$\Rightarrow -b'v - \sum_{i=1}^{n} \log(\exp(0) + \exp(c_i + W_i v))$$

$$= -b'v - \sum_{i=1}^{n} \log(1 + \exp(c_i + W_i v))$$

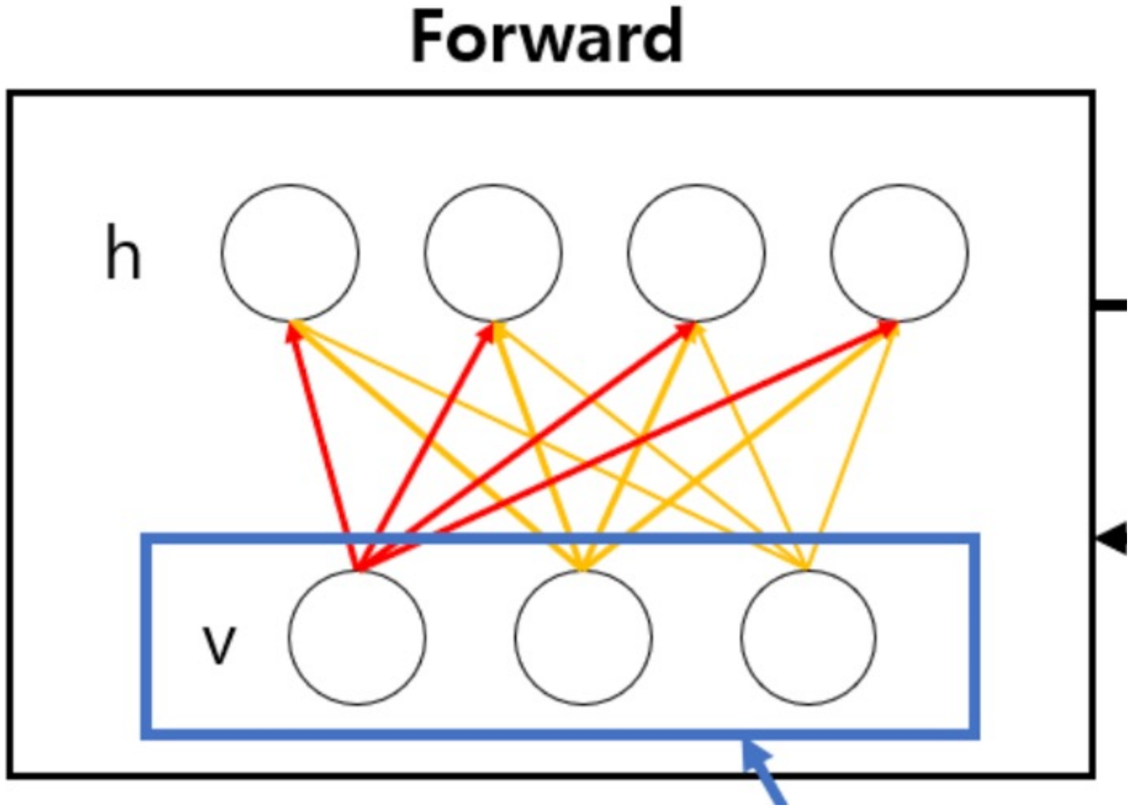


$$\mathbf{W} = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{pmatrix}$$

출처: https://angeloyeo.github.io/2020/10/02/RBM.html

Restricted Boltzmann Machine(RBM)

P(h|v) → Visible 데이터가 존재할 때 hidden unit의 데이터를 샘플링하는 방법



Restricted Boltzmann Machine(RBM)

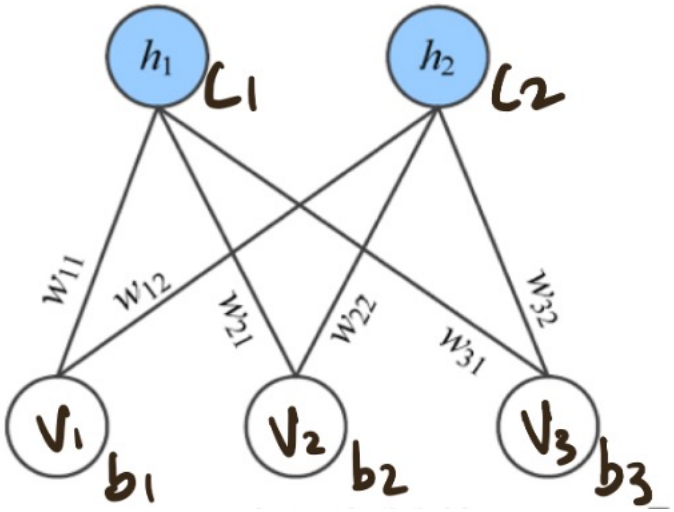
P(h|v) → Visible 데이터가 존재할 때 hidden unit의 데이터를 샘플링하는 방법

$$
\begin{aligned}
p(h|v) &= \frac{p(h,v)}{p(v)} = \frac{\frac{1}{Z}\exp(-E(h,v))}{\sum_h p(h,v)} \\
&= \frac{\frac{1}{Z}\exp(-E(h,v))}{\sum_h \frac{1}{Z}\exp(-E(h,v))} \\
&= \frac{\exp(-E(h,v))}{\sum_h \exp(-E(h,v))} \\
&= \frac{\exp(b'v + c'h + h'Wv)}{\sum_h \exp(b'v + c'h + h'Wv)} \\
&= \frac{\exp(b'v)\exp(c'h + h'Wv)}{\sum_h \exp(b'v)\exp(c'h + h'Wv)} \\
&= \frac{\exp(b'v)\exp(c'h + h'Wv)}{\exp(b'v)\sum_h \exp(c'h + h'Wv)} \\
&= \frac{\exp(c'h + h'Wv)}{\sum_h \exp(c'h + h'Wv)}
\end{aligned}
$$

여기서 총 n개의 hidden layer unit 중 i번 째 hidden layer의 unit이 1로 샘플링될 확률 $p(h_i = 1|v)$를 계산하자.

Bernoulli RBM에서는 $h \in \{0, 1\}$ 이므로,

$$
\begin{aligned}
p(h_i = 1|v) &= \frac{\exp(c_i + W_i v)}{\sum_h \exp(c_i h_i + h_i W_i v)} \\
&= \frac{\exp(c_i + W_i v)}{\exp(0) + \exp(c_i + W_i v)} \\
&= \frac{\exp(c_i + W_i v)}{1 + \exp(c_i + W_i v)} \\
&= \frac{1}{1 + 1/\exp(c_i + W_i v)} \\
&= \sigma(c_i + W_i v)
\end{aligned}
$$



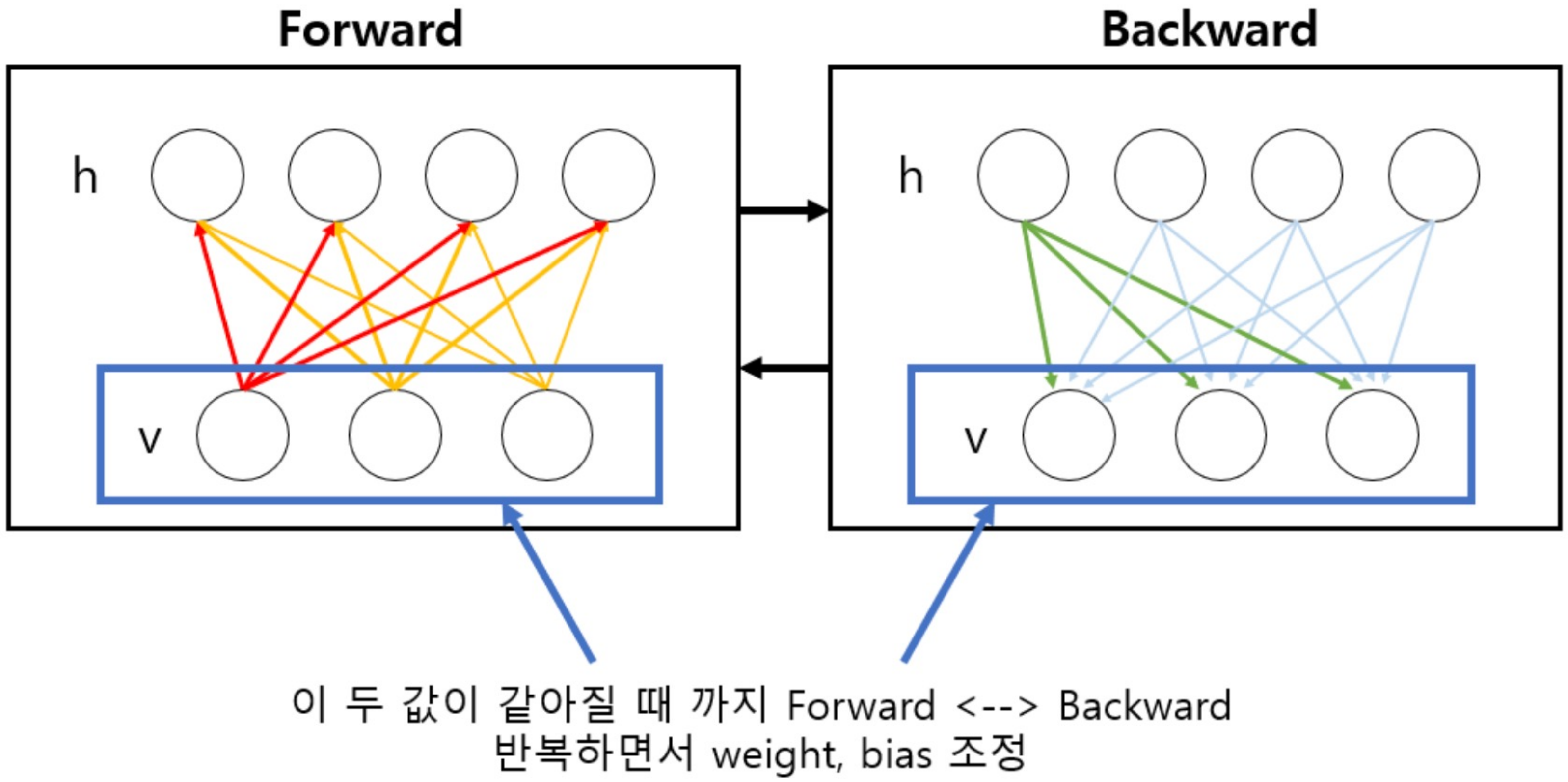
$$p(v_j = 1|h) = \sigma(b_j + W_j' h)$$

주의! RBM에서 sigmoid output은 1이 될 '확률' 즉, deterministic한 값이 아니다

출처: https://angeloyeo.github.io/2020/10/02/RBM.html

Restricted Boltzmann Machine(RBM)

RBM의 학습 과정



출처: https://angeloyeo.github.io/2020/10/02/RBM.html

Restricted Boltzmann Machine(RBM)

RBM의 학습 과정

만약 파라미터 $\theta(W, b, c)$가 잘 학습 됐다면 현재 데이터를 가지고 얻은 가능도(likelihood)의 곱이 최대가 될 것 이다.

P(v|θ)를 θ로 미분해서 최대 값을 찾으면 된다. = negative log likelihood의 최소를 찾으면 된다.

출처: https://angeloyeo.github.io/2020/10/02/RBM.html

Restricted Boltzmann Machine(RBM)

RBM의 학습 과정

$$-\frac{\partial}{\partial\theta}\log p(v) = -\frac{\partial}{\partial\theta}\log\left(\frac{\exp(-F(v))}{Z}\right)$$

$$= -\frac{\partial}{\partial\theta}(\log\exp(-F(v)) - \log(Z))$$

$$= -\frac{\partial}{\partial\theta}(-F(v) - log(Z))$$

$$= \frac{\partial}{\partial\theta}F(v) + \frac{\partial}{\partial\theta}\log(Z)$$

$$= \frac{\partial}{\partial\theta}F(v) + \frac{\partial}{\partial\theta}\log\left(\sum_{\tilde{v}}\exp(-F(\tilde{v}))\right)$$

$$= \frac{\partial}{\partial\theta}F(v) + \frac{\sum_{\tilde{v}}\exp(-F(\tilde{v}))\frac{\partial}{\partial\theta}(-F(\tilde{v}))}{\sum_{\tilde{v}}\exp(-F(\tilde{v}))}$$

$$= \frac{\partial}{\partial\theta}F(v) - \sum_{\tilde{v}}\frac{\exp(-F(\tilde{v}))}{Z}\cdot\frac{\partial}{\partial\theta}(F(\tilde{v}))$$

$$= \frac{\partial}{\partial\theta}F(v) - \sum_{\tilde{v}}p(\tilde{v})\frac{\partial}{\partial\theta}F(\tilde{v})$$

$$\Rightarrow \frac{\partial}{\partial\theta}F(v) - E_p\left\{\frac{\partial F(\tilde{v})}{\partial\theta}\right\}$$

출처: https://angeloyeo.github.io/2020/10/02/RBM.html

Restricted Boltzmann Machine(RBM)

RBM의 학습 과정 – Contrastive Divergence를 줄이자!

$$\Rightarrow \frac{\partial}{\partial\theta}F(v) - E_p\left\{\frac{\partial F(\tilde{v})}{\partial\theta}\right\} \approx \frac{\partial}{\partial\theta}F(v) - \frac{1}{|\mathcal{N}|}\sum_{\tilde{v}\in\mathcal{N}}\frac{\partial F(\tilde{v})}{\partial\theta}$$

Visible data에 대한 free energy와 sampling된 visible data에 대한 free energy의 차이가 loglikelihood를 Maximize 시켜준다.

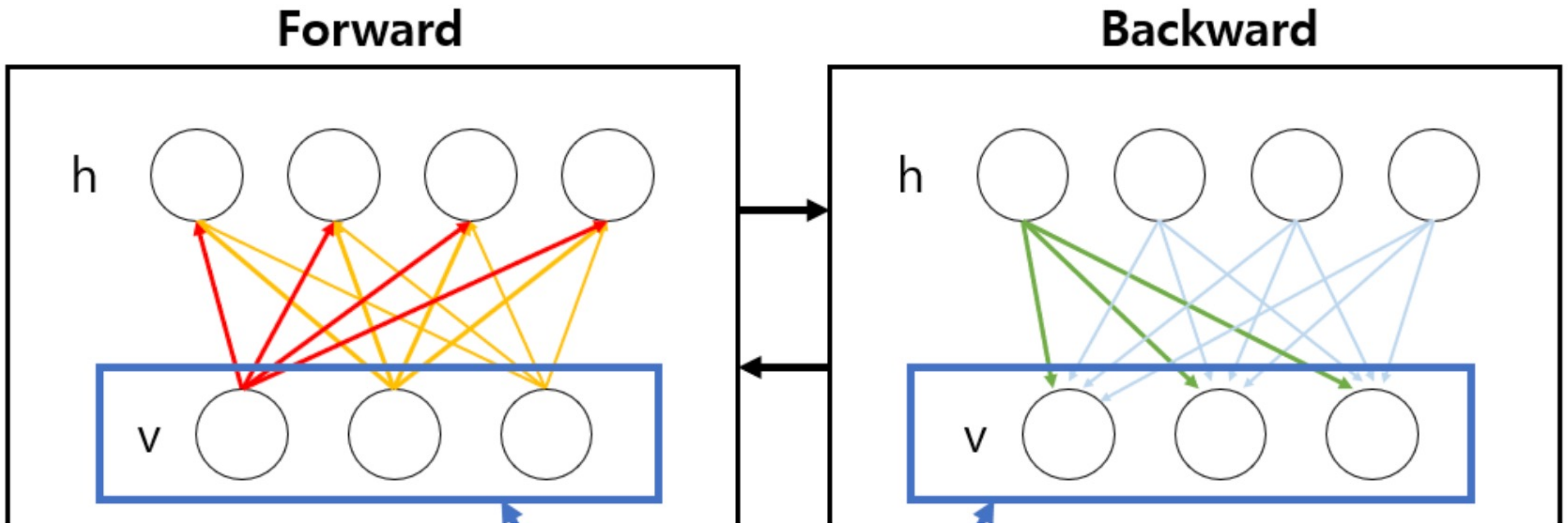
Contrastive Divergence란 Visible data와 sampling된 data 간에 얼마나 차이가 있는지를 나타낸다.
공식 대로라면 sampling을 무한히 많이 수행해야 제대로 된 expectation이 된다.

$$loss = F(v) - F(v^{(1)})$$

출처: https://angeloyeo.github.io/2020/10/02/RBM.html

Restricted Boltzmann Machine(RBM)

RBM의 학습 과정 – Sampling은 어떤 방식으로 진행될까?



Step 1. Sample $\vec{h}^{(l)} \sim P(\vec{h}|\vec{v}^{(l)})$

$\Rightarrow$ We can simultaneously and independently sample from all the elements of $\vec{h}^{(l)}$ given $\vec{v}^{(l)}$.

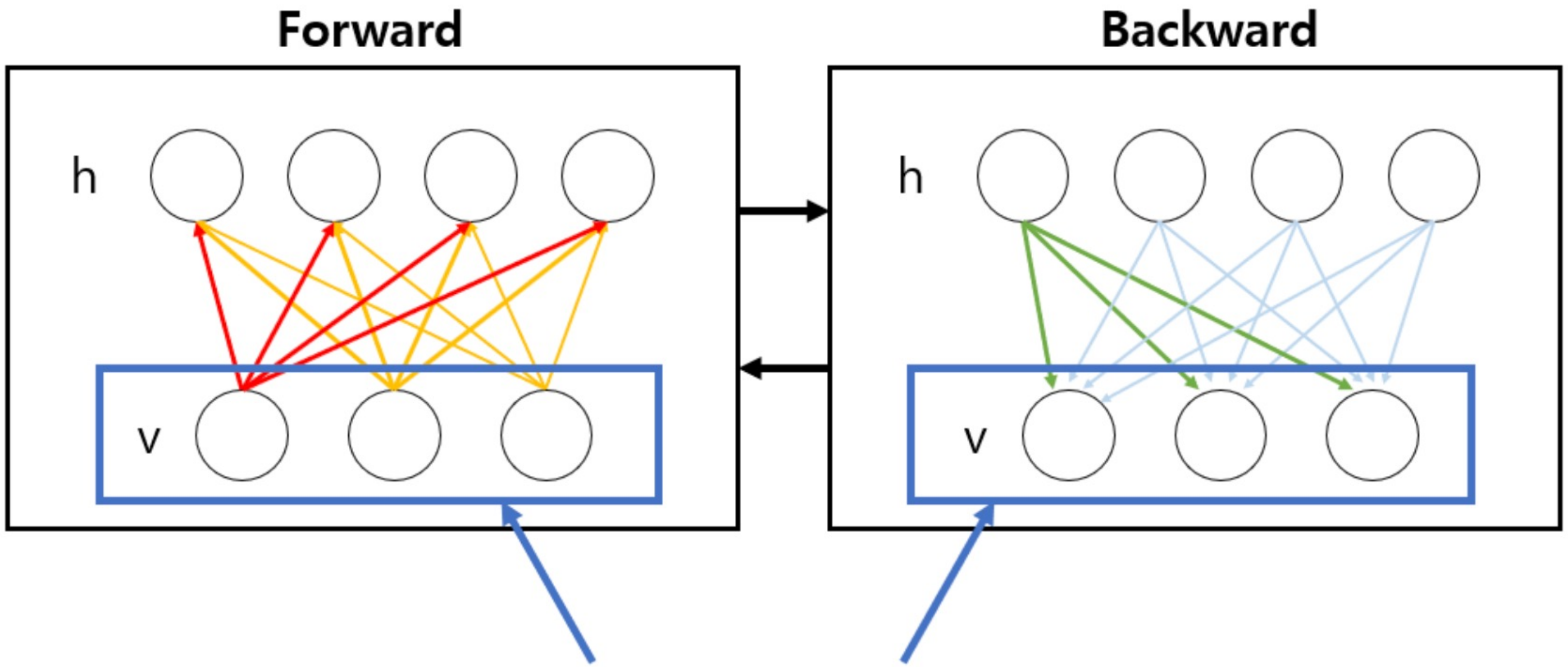
Step 2. Sample $\vec{v}^{(l+1)} \sim P(\vec{v}|\vec{h}^{(l)})$ $\quad p(v_j = 1|h) = \sigma(b_j + W_j' h)$

$\Rightarrow$ We can simultaneously and independently sample from all the elements of $\vec{v}^{(l+1)}$ given $\vec{h}^{(l)}$.

출처: https://angeloyeo.github.io/2020/10/02/RBM.html

Restricted Boltzmann Machine(RBM)

RBM의 학습 과정 – Sampling은 어떤 방식으로 진행될까?



이 두 값이 같아질 때 까지 Forward <--> Backward
반복하면서 weight, bias 조정

num = rand(1);

if num < $P(h_j = 1|\vec{v})$

$\Rightarrow h_j = 1$

else

$\Rightarrow h_j = 0$

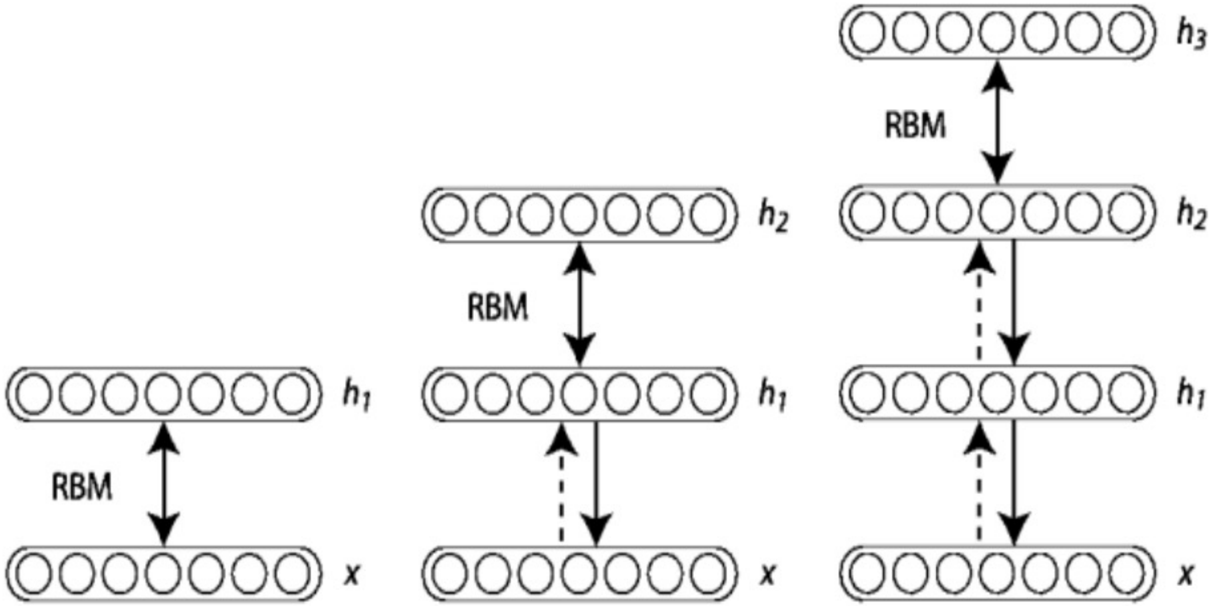
end

출처: https://angeloyeo.github.io/2020/10/02/RBM.html

Deep Belief Network(DBN)

DBN은 RBM을 여러 층 쌓은 모델이다

Pre-training 과정



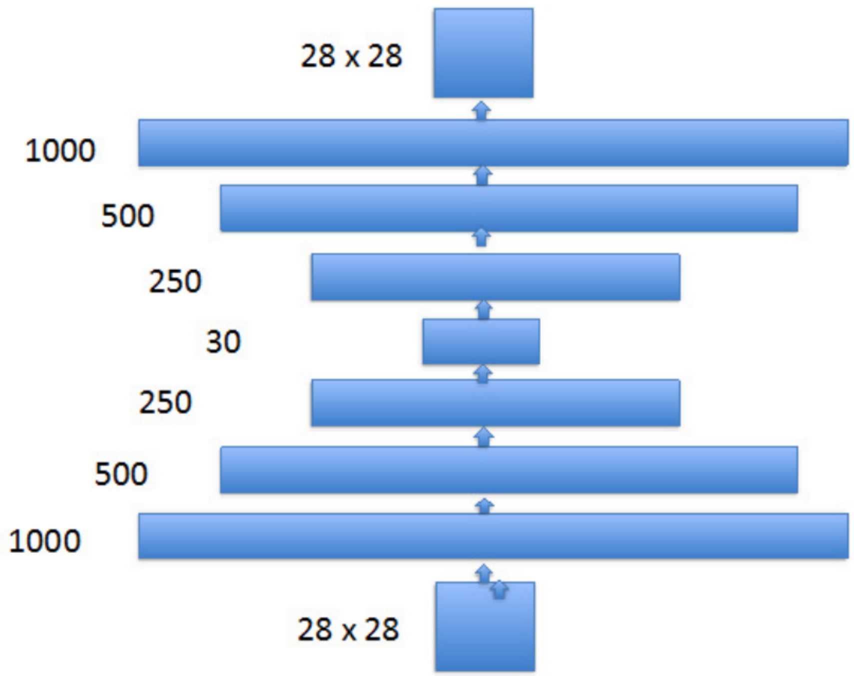
Initialize weight로 사용!

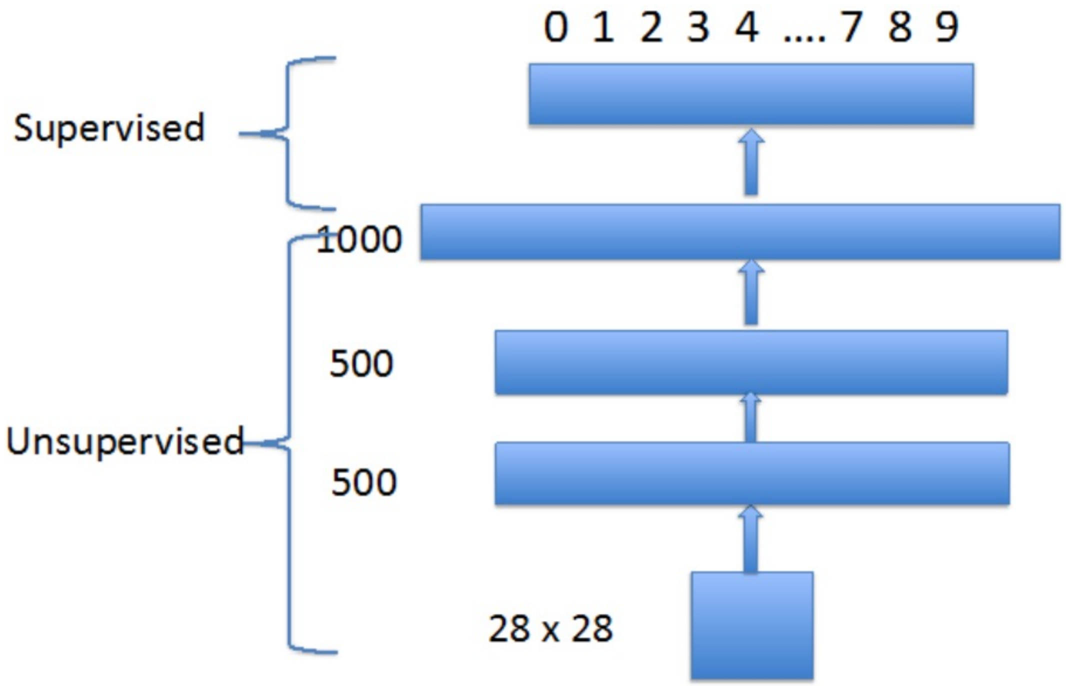
출처: https://ocw.snu.ac.kr/sites/default/files/NOTE/IML_Lecture%20%2809%29.pdf

Deep Belief Network(DBN)

DBN은 RBM을 여러 층 쌓은 모델이다

Pre-training 과정



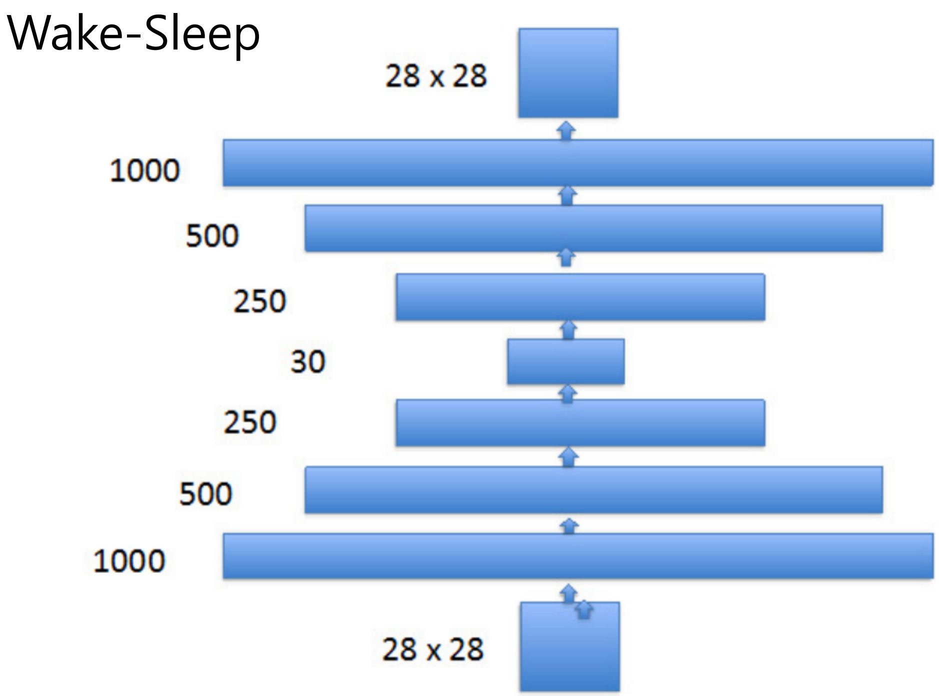


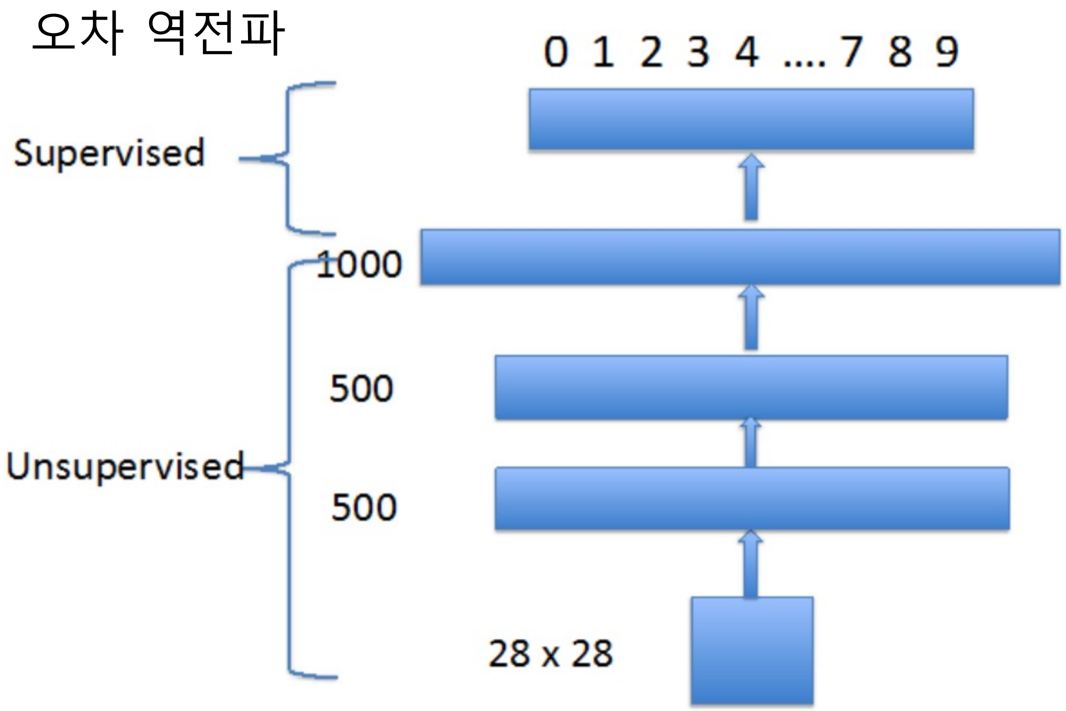
출처: https://ocw.snu.ac.kr/sites/default/files/NOTE/IML_Lecture%20%2809%29.pdf

Deep Belief Network(DBN)

DBN은 RBM을 여러 층 쌓은 모델이다

Fine-Tuning 과정





출처: https://ocw.snu.ac.kr/sites/default/files/NOTE/IML_Lecture%20%2809%29.pdf

Q & A