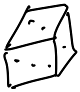
Ch 01. 정규분포

1.1 확률의 기초

1.1.1 확률 변수의 확률 분포

확률 변수 $= x$

 $\rightarrow x = \{1, 2, 3, 4, 5, 6\} \rightarrow$ 표본 or 샘플

$\{\frac{1}{6}, \ldots p(x=3), \ldots, \frac{1}{6}\} \rightarrow$ 확률 분포

모두 같은 확률 $\rightarrow$ 균등 분포

확률 분포가 되기 위한 조건

이산 확률 분포

1. $0 \leq p(x_k) \leq 1 \quad (k=1, \ldots, N)$
2. $\sum_{k=1}^{N} p(x_k) = 1$

연속 확률 분포

1. $p(x) \geq 0$
2. $\int_{-\infty}^{\infty} p(x)\,dx = 1.$

1.1.2 확률 분포의 종류

이산 확률 분포 $\Rightarrow$ x가 이산값 $\Rightarrow$ 

연속 확률 분포 $\Rightarrow$ x가 연속값 $\Rightarrow$ 

1.1.3 기댓값과 분산

기댓값 = 평균

이산 $E[x] = \sum_{x=1}^{N} x_k p(x_k)$

연속 $E[x] = \int_{-\infty}^{\infty} x p(x)\,dx$

분산 = $(x-\mu)^2$의 기댓값

$$Var[x] = E[(x-\mu)^2] = \sum_{k=1}^{N} (x_k-\mu)^2 p(x_k)$$

$$Var[x] = E[(x-\mu)^2] = \int_{-\infty}^{\infty} (x-\mu)^2 p(x)\,dx$$

1.2 정규 분포

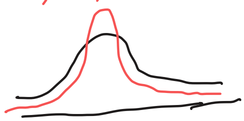
정규 분포 = $p(x;\mu,\sigma) = N(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$

확률 변수 매개변수

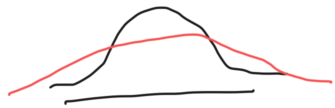
만약 $\mu=0,\ \sigma=1$ 은 $N(x;0,1) = \frac{1}{\sqrt{2\pi}} exp\left(-\frac{x^2}{2}\right)$ ⇒ "표준 정규 분포"

if) μ ⇒ ↑

if) σ ⇒ ↓



if) σ ⇒ ↑



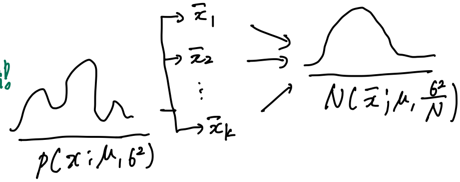
1.3 중심 극한 정리

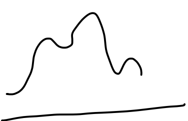
모든 확률 분포 $p(x)$ $\xrightarrow[\text{추출!}]{}$ $\{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$

일수와!

표본 평균 = $\bar{x} = \frac{x^{(1)} + x^{(2)} + \cdots + x^{(N)}}{N}$

즉! →





⇒

if) N=1

if) N=2

. . . if) N=100↑ 더 커져!

1.4 표본합의 확률분포

$$\bar{x} = \frac{x^{(1)} + x^{(2)} + \cdots + x^{(N)}}{N} \overset{\text{표본합}}{\Longrightarrow} S = x^{(1)} + x^{(2)} + \cdots + x^{(N)} = N\bar{x}$$

즉! ① $E[S] = E[N\bar{x}] = NE[\bar{x}] = \boxed{N\mu}$ ☆

② $Var[S] = Var[N\bar{x}] = E[(N\bar{x} - N\mu)^2] = \boxed{N\sigma^2}$

규칙 발견!

$P(x; \mu, \sigma^2)$ 일 때!

$P(\bar{x}; \mu, \frac{\sigma^2}{N})$ 이고!

$P(S; N\mu, N\sigma^2)$ 이다!

이러면 분산이 작아져!

근데! 이러면 평균과 분산 모두 ↑

Ch 02. 최대 가능도 추정

1. 생성 모델 개요
2. 실제 데이터로 생성 모델 구현하기
3. 최대 가능도 추정 이론
4. 생성 모델의 용도

2.1 생성 모델 개요

2.1.1 생성 모델의 목표

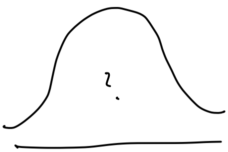
$p(x; ?, ?)$

$\xrightarrow{\text{샘플링}} (x^{(1)}, x^{(2)}, \cdots, x^{(N)})$

어디서 왔지?

↳ 이걸 찾는 게 생성 모델의 목표! ⇒ 이러한 과정을 추론(inference) 혹은 매개변수 추정(parameter estimation)

2.1.2 모집단과 샘플



모집단 분포! (population distribution)
전체 집합!

$p_*(x)$ ← 너무 알고 싶어 ㅠㅠ 그럼, 아래!! 처럼!

모델링! → 1. $p_*(x)$가 뭔지 몰라. 그러면 어떤 매개변수로 표현되는 분포라고 가정하자!

매개변수 추정! → 2. 그리고 $p_*(x)$에서 얻은 샘플들이 $p_*(x)$에 맞게 매개변수를 조정하자!

2.3 최대 가능도 추정 이론

$\xrightarrow{\text{샘플링}} \{x^{(1)}, x^{(2)}, \cdots, x^{(N)}\}$

이때!

1. 모델링부터 → 정규 분포라고 하자! → 그럼 매개변수는 μ, σ!

2. 매개변수 추정을 하자 → 그러면!

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} x^{(n)}$$

$$\hat{\sigma} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} (x^{(n)} - \hat{\mu})^2}$$

왜인지는 뒤에서!

2.3.1 가능도 최대화

→ 이때! 매개변수 = θ 라면, 데이터 x을 얻을 확률밀도 = $p(x;\theta)$

그럼!

$\xrightarrow{\text{샘플링}}$ $D = \{x^{(1)}, x^{(2)}, \cdots, x^{(N)}\}$ 일 때!

이걸 얻을 확률 밀도 $= p(D;\theta) = p(x^{(1)};\theta)\, p(x^{(2)};\theta) \cdots p(x^{(N)};\theta)$

$$= \prod_{n=1}^{N} p(x^{(n)};\theta)$$

매개변수 = θ 일 때, D를 얻을 확률이닷!

이걸, θ를 인수로 받는 함수!라고 하면! 아래처럼 표현!

$L(\theta) = p(D;\theta)$ = 가능도 (likelihood) ☆중요☆

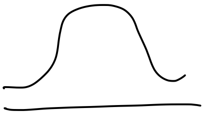
= 가능도 함수 (likelihood function)

여기서*

$\log L(\theta) = \log p(D;\theta)$ = 로그 가능도☆

자몽 여기서 가능도의 최대화란?
θ를 잘 찾아서?
D가 딱! 맞는 분포를 찾는거지?
즉! D를 얻을 확률을
'최대화' 하는 것이로다!

2.3.2 정규 분포의 최대 가능도 추정



$$\mathcal{N}(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

$$\xrightarrow{\text{샘플링}} D = \{x^{(1)}, x^{(2)}, \cdots, x^{(N)}\} \xrightarrow{\text{확률밀도}} \underbrace{p(D;\mu,\sigma) = \prod_{n=1}^{N} p(x^{(n)};\mu,\sigma)}_{\text{이게 가능도!}}$$

$$\log p(D;\mu,\sigma) = \log \prod_{n=1}^{N} p(x^{(n)};\mu,\sigma)$$

$$= \underbrace{-\frac{N}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{n=1}^{N}(x^{(n)}-\mu)^2}_{\text{이게 로그 가능도!}}$$

자! 여기서! 최대 가능도를 어떻게 얻을까!

→ 어쨌든 $\log p(D;\mu,\sigma)$가 최대화 되는 μ, σ를 찾아야 해

→ 2차 미분?

→ 2차 미분값 = 0 이 되는 곳?

→ 아... 2차 조건이... 2차 함수.. 2차 계수가 ⊖인...

→ 오? 로그 가능도를 보니?....!!!

→ 찾았드아!!!

결론: μ와 σ로 편미분 하면!!! → 아래식! → 오! 증명됐드아!!!

$$\frac{\partial \log p(D;\mu,\sigma)}{\partial \mu} = 0$$

$$\Leftrightarrow \mu = \frac{1}{N}\sum_{n=1}^{N} x^{(n)}$$

$$= \hat{\mu}$$

$$\frac{\partial \log p(D;\mu,\sigma)}{\partial \sigma} = \frac{\partial \log p(D;\mu=\hat{\mu},\sigma)}{\partial \sigma}$$

$$\Leftrightarrow \sigma = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x^{(n)}-\hat{\mu})^2}$$

$$= \hat{\sigma}$$

Ch 03. 다변량 정규 분포

1. 넘파이와 다차원 배열

2. 다변량 정규 분포

3. 2차원 정규 분포 시각화

4. 다변량 정규 분포의 최대 가능도 추정

3.1 넘파이의 다차원 배열

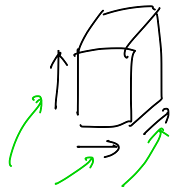
3.1.1 다차원 배열

$\circ$ → 스칼라 → 0차원 텐서

$\begin{pmatrix}\circ\\\circ\\\circ\end{pmatrix}$ → 벡터 → 1차원 텐서 → 모두 텐서!

$\begin{pmatrix}\circ&\circ\\\circ&\circ\end{pmatrix}$ → 행렬 → 2차원 텐서

이때!



축! axis → 3차원 배열

3.1.3 원소별 연산

$\begin{pmatrix}\circ&\circ\\\circ&\circ\end{pmatrix}+\begin{pmatrix}\circ&\circ\\\circ&\circ\end{pmatrix}=\begin{pmatrix}\circ&\circ\\\circ&\circ\end{pmatrix}$ ⇒ 원소별 합

$\begin{pmatrix}\circ&\circ\\\circ&\circ\end{pmatrix}\times\begin{pmatrix}\circ&\circ\\\circ&\circ\end{pmatrix}=\begin{pmatrix}\circ&\circ\\\circ&\circ\end{pmatrix}$ ⇒ 원소별 곱 ⇒ 아다마르 곱 (Hadamard product)

3.1.4 벡터의 내적과 행렬 곱

$\begin{pmatrix}\circ\\\circ\\\circ\end{pmatrix}\cdot\begin{pmatrix}\circ\\\circ\\\circ\end{pmatrix}=\triangle+\triangle+\triangle$ ⇒ 벡터의 내적

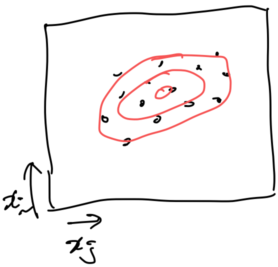
$\begin{pmatrix}\circ&\circ\\\circ&\circ\end{pmatrix}\cdot\begin{pmatrix}\circ&\circ\\\circ&\circ\end{pmatrix}=\triangle\triangle+\triangle\triangle+\triangle\triangle+\triangle\triangle$ ⇒ 행렬곱

3.2 다변량 정규 분포

3.2.1 다변량 정규 분포 공식

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix} \Rightarrow N(x;\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^p |\Sigma|}} \exp\left\{ -\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu) \right\}$$

도식화



평균벡터

$$= \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{pmatrix}$$

공분산 행렬

$$= \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & & \\ \vdots & & \ddots & \\ \sigma_{p1} & & & \sigma_{pp} \end{pmatrix}$$

여기서!

$\sigma_{11}, \sigma_{22} \cdots \sigma_{pp}$ → 분산! ⇒ x_i가 얼마나 흩어졌는지 보여줘!

$\sigma_{12}, \sigma_{21}, \cdots$ → 공분산! ⇒ x_i와 x_j의 연관성을 보여줘!

2설. 공분산은 어떻게 계산?

분산 $= Var[x_i] = E[(x_i-\mu_i)^2]$ $\overset{\text{공분산!}}{\Longleftrightarrow}$ 공분산 $= Cov[x_i, x_j] = E[(x_i-\mu_i)(x_j-\mu_j)]$

요건 외워!

2절. 전치는?

$x = (1,2,3) \overset{\text{전치!}}{\Longrightarrow} x^T = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$

2절. 행렬식은?

이건 알지? determinant!

2절. 역행렬은?

$AA^{-1} = A^{-1}A = I$ ⇒ inverse matrix!

3.4 다변량 정규 분포의 최대 가능도 추정

3.4.1 최대 가능도 추정하기

$$N(x;\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^D|\Sigma|}} \exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\}$$

요기서! 샘플 $D = \{x^{(1)}, x^{(2)}, \cdots x^{(N)}\}$ 이면! 요 'D'가 샘플링될 확률은!

$$p(D;\mu,\Sigma) = N(x^{(1)};\mu,\Sigma)N(x^{(2)};\mu,\Sigma)\cdots N(x^{(N)};\mu,\Sigma)$$

$$= \prod_{n=1}^{N} N(x^{(n)};\mu,\Sigma)$$

요건 2장과 같아! → 즉! μ, Σ가 매개변수 일 때, 샘플 D를 얻은 확률 밀도!

따라서!

$p(D;\mu,\Sigma) = L(\mu,\Sigma)$ = 가능도! = 가능도 함수당!!

혹은! ↳ $\log p(D;\mu,\Sigma)$

여기로 마찬가지! 최대 가능도

$$\frac{\partial L}{\partial \mu} = 0 \Rightarrow \hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} x^{(n)}$$

$$\frac{\partial L}{\partial \Sigma} = 0 \Rightarrow \hat{\Sigma} = \frac{1}{N}\sum_{n=1}^{N}(x^{(n)}-\hat{\mu})(x^{(n)}-\hat{\mu})^T$$

Ch 04. 가우스 혼합 모델

1. 우리 주변의 다봉 분포

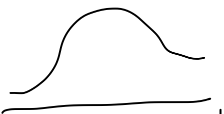
2. 가우스 혼합 모델 데이터 생성

3. 가우스 혼합 모델의 수식

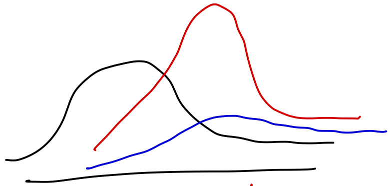
4. 매개변수 추정의 어려움

4.1 우리 주변의 다봉 분포

가우스 분포



하나면 '정규 분포'



여러 개면? → 가우스 혼합 모델 (Gaussian Mixture Model, GMM)

4.2 가우스 혼합 모델 데이터 생성

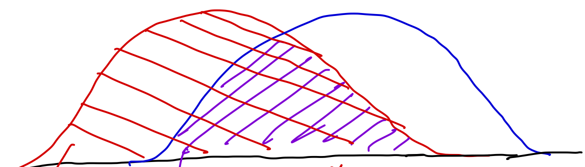
모든 다봉 분포 $\xrightarrow{\text{!WoW!}}$ GMM으로 표현 가능! $\xrightarrow{\text{근데!}}$ 2가지 작업이 필요!

1! 모델링 : 관측 데이터의 분포를 GMM으로 표현할 수 있다고 가정!

2! 매개변수 추정 : GMM의 매개 변수 추정

↳ 근데! 이건 최대 가능도로 안돼 ㅠㅠ → 그래서! EM!을 사용

4.3 가우스 혼합 모델의 수식



결합 확률 : 동시에 발생!
(Joint probability, $p(x,y)$)

주변확률 : 개별 사건 발생 $\xrightarrow{\text{결합에서 변형가능!}}$ $p(x) = \sum_y p(x,y)$ or $\int p(x,y)\,dy$
(marginal probability, $p(x)$ or $p(y)$)

근데! y가 이미 일어난 상황에서 x의 발생 확률은?!

⇒ 조건부 확률 → 특정 조건 필요!
(Conditional probability)

곱셈 정리!

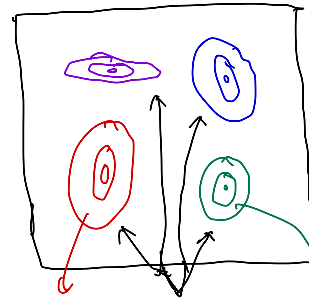
↳ $p(x,y) = p(x|y)\,p(y)$

GMM의 데이터 생성 절차!

→ 우선 K개의 정규 분포를 준비!

그리고 반복!

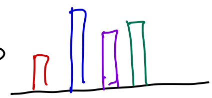
1. 특정 확률 분포에 따라 K개의 정규분포 중 하나 선택!
2. 선택한 정규 분포에서 데이터 생성!



$D=\{x^1,\cdots,x^n\}$ 하나 선택! 그리고 샘플링! $D=\{x^1,\cdots,x^n\}$

여기서!

↳ 이걸 범주형 분포로 표현! (categorical distribution) →



↓

$p(z=k;\phi)=\phi_k$

따라서 최종! GMM 수식은!?!?

1. K개 정규 분포 중 z 값에 따라 정규분포 선택! 즉! z=k라면, x가 따르는 확률 분포는 k번째!

따라서! → $p(x|z=k;\mu,\Sigma) = N(x;\mu_k,\Sigma_k)$

2. 생성 모델의 목표! → 관측 데이터 x의 확률 분포 $p(x)$를 표현

따라서 → 주변화 사용! → $p(x)=\sum_{k=1}^{K} p(x,z=k)$

3. 이때, 결합 확률은? 곱·셈·정·리

→ $p(x,z=k)=p(z=k)\,p(x|z=k)=\phi_k N(x;\mu_k,\Sigma_k)$

4. 따라서 ㅠㅠ 결국 ㅠㅠ

→ $p(x)=\sum_{k=1}^{K} \phi_k N(x;\mu_k,\Sigma_k)$

4.4 매개변수 추정의 어려움

GMM은 매개변수 추정이 어려워..~

$p(x;\underline{\phi,\mu,\Sigma}) = \sum_{k=1}^{K} \phi_k N(x;\mu_k,\Sigma_k)$ 인데 ~ $D=\{x^{(1)}, x^{(2)}, \cdots, x^{(N)}\}$을 얻었다면 ~

$p(D;\theta) = p(x^{(1)};\theta)p(x^{(2)};\theta)\cdots p(x^{(N)};\theta) = \prod_{n=1}^{N} p(x^{(n)};\theta)$ ⟹ 가능도!

$$\log p(D;\theta) = \log \prod_{n=1}^{N} p(x^{(n)};\theta)$$

$$= \sum_{n=1}^{N} \log p(x^{(n)};\theta)$$

$$= \sum_{n=1}^{N} \log\left(\sum_{k=1}^{K} \phi_k N(x^{(n)};\mu_k,\Sigma_k)\right)$$ ⟹ 로그 가능도

로그-합! → 분석·해석 어려워 ㅠㅠ
(log-sum) 편미분! 어려워 ㅠㅠ

합-로그! → 해석적으로 쉬워 등등

즉! $\log p(D;\theta)$으로는 최대 가능도 어려워 ㅠㅠ

Ch 05. EM 알고리즘

1. KL 발산
2. EM 알고리즘 도출 ①
3. EM 알고리즘 도출 ②
4. GMM과 EM 알고리즘
5. EM 알고리즘 구현

5.1 KL 발산

우선 표기법을 변경!

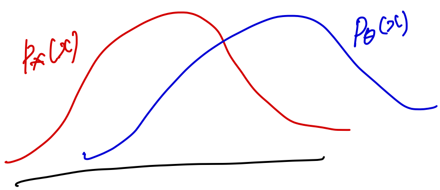
$E_{p(x)}[f(x)] = \int f(x)p(x)\,dx$ ⇒ 연속 확률 변수 x가 있고 그 확률분포가 $p(x)$ 일 때,
함수 $f(x)$의 기댓값을 의미!

KL 발산 (쿨백-라이블러 발산, Kullback-Leibler divergence) 란?!?!
⇒ 두 확률 분포의 차이!를 측정하는 척도!

$$\Rightarrow D_{KL}(p\|q) = \int p(x) \log \frac{p(x)}{q(x)}\,dx \quad \text{또는!} \quad \sum_x p(x) \log \frac{p(x)}{q(x)}$$

KL 발산의 특징은?
1. 두 확률분포가 다를수록 값이 커진다!
2. 두 확률분포가 같으면 0이다!
3. 같지 않으면 0보다 크다!
4. $D_{KL}(p\|q)$와 $D_{KL}(q\|p)$는 다르다!

KL 발산과 최대 가능도 추정의 관계



$p_*(x) \xrightarrow{\text{샘플링}} \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$ ⇒ 이걸로 $p_\theta(x)$ 최대가능도!

즉! $\log \prod_{n=1}^{N} p_\theta(x^{(n)}) = \sum_{n=1}^{N} \log p_\theta(x^{(n)})$

따라서! ⇒ $\hat{\theta} = \arg\max_\theta \sum_{n=1}^{N} \log p_\theta(x^{(n)})$

핵심 아이디어는!

$p_\theta(x)$를 $p_*(x)$에 최대한 가깝게 만든다

⇕

$p_*(x)$와 $p_\theta(x)$의 KL 발산을 최소화한다

즉! $D_{KL}(p_*\|p_\theta) = \int p_*(x) \log \frac{p_*(x)}{p_\theta(x)}\,dx$를 계산!

그러나! $p_*(x)$를 정확하게 모른다...
따라서! 몬테카를로 방법 (Monte Carlo method)을 사용

몬테카를로 방법은?

=> 확률분포나 기댓값의 근사값을 구하는 기법!

=> 무작위 샘플 기반의 결과값의 평균을 구해서 근사!

=> 근사 방법은?

1. $p_*(x)$에 따라 샘플 $\{x^{(1)}, x^{(2)}, \dots, x^{(N)}\}$을 생성!

2. 각 데이터 $x^{(i)}$에서 $f(x^{(i)})$를 구하고 평균을 구한다

즉! $$E_{p_*(x)}[f(x)] = \int p_*(x) f(x)\,dx$$

$$\approx \frac{1}{N}\sum_{n=1}^{N} f(x^{(n)}) \quad (x^{(n)} \sim p_*(x))$$

몬테카를로를 KL발산에 적용하면!

$$D_{KL}(p_* \| p_\theta) = \int p_*(x) \log \frac{p_*(x)}{p_\theta(x)}\,dx$$

$$\approx \frac{1}{N}\sum_{n=1}^{N} \log \frac{p_*(x^{(n)})}{p_\theta(x^{(n)})} \quad (x^{(n)} \sim p_*(x))$$

$$= \frac{1}{N}\sum_{n=1}^{N} \left(\log p_*(x^{(n)}) - \log p_\theta(x^{(n)})\right)$$

여기서!, 가장 작은 KL은?

$$\underset{\theta}{\arg\min}\, D_{KL}(p_* \| p_\theta)$$

$$\approx \underset{\theta}{\arg\min} \left(-\frac{1}{N}\sum_{n=1}^{N} \log p_\theta(x^{(n)})\right)$$

$$= \underset{\theta}{\arg\max} \sum_{n=1}^{N} \log p_\theta(x^{(n)})$$

즉! $$\underbrace{\underset{\theta}{\arg\min}\, D_{KL}(p_* \| p_\theta)}_{\text{최소 KL발산!}} \approx \underbrace{\underset{\theta}{\arg\max} \sum_{n=1}^{N} \log p_\theta(x_n)}_{\text{로그 최대가능도!}}$$

↳ 최소 KL발산!

↳ 로그! 최대가능도!

5.2 EM 알고리즘 도출 ①

모델 + 잠재변수 → 확률의 주변화 → 로그 가능도

관찰 가능한 확률 변수

잠재변수

매개변수

여러 개의 데이터 $D=\{x^{(1)}, x^{(2)}, \dots x^{(N)}\}$

하나의 데이터 → $\log p_\theta(x) = \log \sum_z p_\theta(x, z)$

$$\log p_\theta(D) = \log\left(p_\theta(x^{(1)})\, p_\theta(x^{(2)}) \cdots p_\theta(x^{(N)})\right)$$

$$= \sum_{n=1}^{N} \log p_\theta(x^{(n)})$$

$$= \sum_{n=1}^{N} \log \sum_{z^{(n)}} p_\theta(x^{(n)}, z^{(n)})$$

이때! "로그-합"이라서 어려워!

이를 위해, 임의 확률 $q(z)$를 사용

그러면!? 1단계!

$$\log p_\theta(x) = \log \frac{p_\theta(x, z)}{p_\theta(z|x)}$$

$$= \log \frac{p_\theta(x, z)\, q(z)}{p_\theta(z|x)\, q(z)}$$

$$= \underbrace{\log \frac{p_\theta(x, z)}{q(z)} + \log \frac{q(z)}{p_\theta(z|x)}}$$

아직 로그-합이야!

그래서 이걸 KL발산으로 바꿔!

⇒ 그러면 2단계!

$$\log p_\theta(x)$$

$$= \log p_\theta(x) \sum_z q(z)$$

$$= \sum_z q(z) \log p_\theta(x)$$

$$= \sum_z q(z) \left(\log \frac{p_\theta(x, z)}{q(z)} + \log \frac{q(z)}{p_\theta(z|x)} \right)$$

이건 KL발산!

$$= \sum_z q(z) \log \frac{p_\theta(x, z)}{q(z)} + \sum_z q(z) \log \frac{q(z)}{p_\theta(z|x)}$$

$$= \underbrace{\sum_z q(z) \log \frac{p_\theta(x, z)}{q(z)} + D_{KL}(q(z), p_\theta(z|x))}$$

이러면 합-로그!

즉! 극복!

5.3 EM 알고리즘 도출 ②

ELBO (Evidence Lower BOund, 증거하한) 은! → { 가장 낮은 로그 가능도!
에이.. 아무리 낮아도 이것보다 가능도가 낮을까..
요런 느낌! }

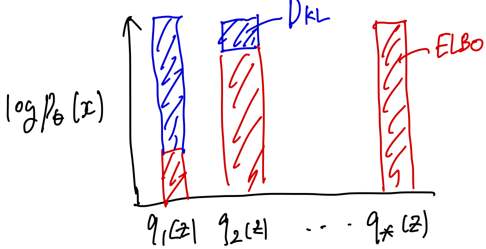
즉!

$$\log p_\theta(x) = \sum_z q(z)\log\frac{p_\theta(x,z)}{q(z)} + \underbrace{D_{KL}(q(z)\,\|\,p_\theta(z|x))}_{\geq 0}$$

$$\geq \sum_z q(z)\log\frac{p_\theta(x,z)}{q(z)} \Longrightarrow$$ 요것이! ELBO!!! ⟹ 즉! $$ELBO(x;q,\theta) = \sum_z q(z)\log\frac{p_\theta(x,z)}{q(z)}$$

자 그럼

$$\log p_\theta(x) = ELBO(x;q,\theta) + D_{KL}(q(z)\,\|\,p_\theta(z|x))$$ $\xrightarrow[\text{즉!}]{\log p_\theta(x)\text{는 일정}}$ ELBO ↑ DKL ↓ or ELBO ↓ DKL ↑



즉! ⟹ DKL 최소화 = ELBO 최대화 = 최대가능도!

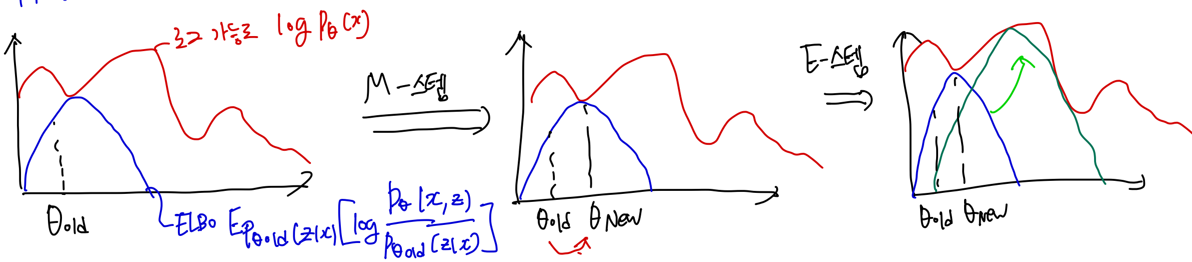
그럼! 우선 $q(z)$ 최적화! ⟹ $\theta=\theta_{old}$로 고정 $\xrightarrow{D_{KL}=0}$ $p_{\theta_{old}}(z|x) = q(z)$

↳ $$ELBO(x;q=p_{\theta_{old}}(z|x),\theta) = \sum_z p_{\theta_{old}}(z|x)\log\frac{p_\theta(x,z)}{p_{\theta_{old}}(z|x)}$$

이것이 E-스텝!! (Expectation value-step) ⟹ $$= E_{p_{\theta_{old}}(z|x)}\left[\log\frac{p_\theta(x,z)}{p_{\theta_{old}}(z|x)}\right]$$

그럼! 다음은 θ 최적화 ⟹ θ에 대한 미분! ⟹ 최적 θ 발견! ⟹ 이것이 M-스텝 (Maximization-step)

이후 E-스텝, M-스텝 반복.. 즉!



2절 데이터가 여러 개일 때!

관측 데이터 $= \{x^{(1)}, x^{(2)}, \cdots, x^{(N)}\}$, 임의 확률 분포 $= \{q^{(1)}, q^{(2)}, \cdots, q^{(N)}\}$ 일 때!

2절! 로그 가능도와 ELBO의 관계는!

$$\sum_{n=1}^{N} \log \sum_{z^{(n)}} p_\theta(x^{(n)}, z^{(n)}) \geq \sum_{n=1}^{N} \mathrm{ELBO}(x^{(n)}; q^{(n)}, \theta)$$
$$= \sum_{n=1}^{N} \sum_{z^{(n)}} q^{(n)}(z^{(n)}) \log \frac{p_\theta(x^{(n)}, z^{(n)})}{q^{(n)}(z^{(n)})}$$

EM 알고리즘은!

1. E-스텝 : $\{q^{(1)}, q^{(2)}, \cdots, q^{(N)}\}$ 갱신 (θ 고정)
   → 각 n에 대해 $q^{(n)}(z) = p_\theta(z \mid x^{(n)})$으로 정함.
2. M-스텝 : θ 갱신 ($\{q^{(1)}, q^{(2)}, \cdots, q^{(N)}\}$ 고정)
   → $\sum_{n=1}^{N} \mathrm{ELBO}(x^{(n)}; q^{(n)}, \theta)$가 최대가 되는 θ 갱신
3. 종료 : 이전 로그 가능도와 이후 로그 가능도 비교

즉! GMM에 대한 EM은?

E-스텝은 각 n, k에 대해

$$q^{(n)}(k) = \frac{\phi_k \mathcal{N}(x^{(n)}; \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \phi_j \mathcal{N}(x^{(n)}; \mu_j, \Sigma_j)}$$

→ 전체 GMM 중 k 파라미터 분포가 같힐 확률?!?!

종료 판정은

$$\underbrace{\frac{1}{N} \sum_{n=1}^{N} \overbrace{\log \underbrace{\sum_{j=1}^{K} \phi_j \mathcal{N}(x^{(n)}; \mu_j, \Sigma_j)}_{\text{GMM 수식}}}^{\text{GMM 로그 가능도}}}_{\text{GMM 평균 로그 가능도}}$$

M-스텝은 각 k에 대해

$$\phi_k = \frac{1}{N} \sum_{n=1}^{N} q^{(n)}(k)$$

$$\mu_k = \frac{\sum_{n=1}^{N} q^{(n)}(k)\, x^{(n)}}{\sum_{n=1}^{N} q^{(n)}(k)}$$

$$\Sigma_k = \frac{\sum_{n=1}^{N} \left(q^{(n)}(k) (x^{(n)} - \mu_k)(x^{(n)} - \mu_k)^T\right)}{\sum_{n=1}^{N} q^{(n)}(k)}$$

⟹ 모든 데이터에 대한 계산!
k 번째 파라미터 기반으로!
전체 데이터에 대입?!?!
이후, 가장 적절한 파라미터 선택?!?!

Ch 06. 신경망

1. 파이토치와 경사법
2. 선형 회귀
3. 매개변수와 옵티마이저
4. 신경망 구현
5. 토치 비전과 데이터셋

6.2 선형 회귀

y와 x가 선형 ⇒ $\hat{y} = Wx + b$

이때! $y-\hat{y}$ 가 정규분포를 따른다고 가정

⇒ $y-\hat{y}$이 평균 0, 표준편차 σ를 따른다고 가정

즉! y는 평균 $\hat{y}$, 표준편차 σ인 정규분포를 따른다!

이는!

$$\hat{y} = Wx + b$$

$$p(y|x;W,b) = N(y;\hat{y},\sigma)$$

따라서 로그 가능도는!

$$\log p(y|x;W,b)$$

$$= \log N(y;\hat{y},b)$$

$$= -\frac{1}{2\sigma^2}(y-Wx-b)^2 + \log\frac{1}{\sqrt{2\pi\sigma^2}}$$

최대 로그 가능도는!

$$\underset{W,b}{\operatorname{argmax}} \log p(y|x;W,b)$$

$$= \underset{W,b}{\operatorname{argmax}} \left(-\frac{1}{2\sigma^2}(y-Wx-b)^2\right)$$

$$= \underset{W,b}{\operatorname{argmin}} \left(y-(Wx+b)\right)^2$$

오차 최소화 함수 (손실 함수)는!

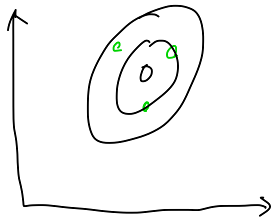
$$L(W,b) = \frac{1}{N}\sum_{n=1}^{N}\left(y^{(n)} - (Wx^{(n)}+b)\right)^2$$

이것이! 평균 제곱 오차 (mean squared error, mse)!

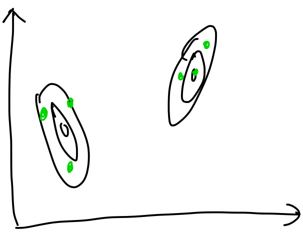
Ch 07. 변이형 오토인코더

1. VAE와 디코더
2. VAE와 인코더
3. ELBO 최적화
4. VAE 구현

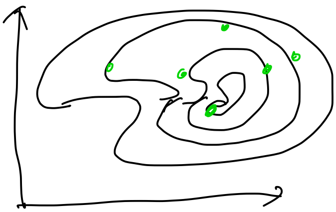
17.1 VAE와 디코더



하나의 정규 분포



GMM



VAE

하나의 정규분포는!

매개변수 $= \theta = \{\mu, \Sigma\}$, 데이터 $= x$ $\Rightarrow$ 확률 분포 $= p_\theta(x) = N(x; \theta)$ 이고!

N개 관측데이터 $D = \{x^{(1)}, x^{(2)}, \cdots, x^{(N)}\} \xrightarrow{\text{로그 가능도}} \log p_\theta(D) = \log\left(p_\theta(x^{(1)})\, p_\theta(x^{(2)}) \cdots p_\theta(x^{(N)})\right)$

GMM은!

1. 범주형 분포에 따라 K개의 정규분포 중 하나를 선택
2. 선택한 정규분포에서 데이터를 생성

$\Rightarrow$ ELBO를 목적함수로! $\Longrightarrow$ EM 알고리즘 수행!

이때! ELBO는

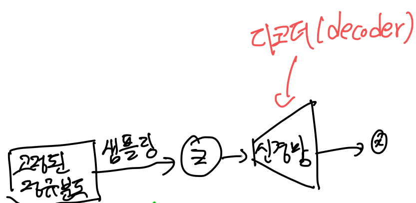
$$\log p_\theta(D) \geq \sum_{n=1}^{N} \sum_{z^{(n)}} q^{(n)}(z^{(n)}) \log \frac{p_\theta(x^{(n)}, z^{(n)})}{q^{(n)}(z^{(n)})}$$

VAE는?

VAE의 데이터 생성 과정

1. 잠재 변수 z를 고정된 정규 분포에서 생성!
2. 신경망을 통해 z를 x로 변환!

즉!



따라서!

목표 : 관측 변수 x의 확률분포 $p(x)$를 얻기!

이를 위해 신경망 $p(x|z)$를 모델링!

$(z \rightarrow x)$

따라서!

$$\underbrace{\hat{x}}_{\text{출력}} = \text{NeuralNet}(\underbrace{z}_{\text{입력}}; \underbrace{\theta}_{\text{신경망의 매개변수}})$$

$p_\theta(x|z) = N(x; \hat{x}, I)$ $\Longrightarrow$ x는 $\hat{x}$을 평균 벡터로 삼는 정규분포를 따른다!

VAE의 디코더의 수행 순서

$$\boxed{\mathcal{N}(z;0,I)} \xrightarrow{\text{샘플링}} z \xrightarrow{\text{입력}} \boxed{\text{디코더}} \xrightarrow{\text{출력}} \hat{x} \rightarrow \boxed{\mathcal{N}(x;\hat{x},I)} \dashrightarrow x$$

그럼 EM으로 할 수 있나? → NO!

VAE에서 E-스텝 → $q^{(n)}(z) = p_\theta(z|x^{(n)}) = \dfrac{p_\theta(x^{(n)}, z)}{p_\theta(x^{(n)})}$

$$= \frac{p_\theta(x^{(n)}, z)}{\int p_\theta(x^{(n)}, z)\,dz}$$

GMM : 이산 → Σ → 계산량 ↓

VAE : 연속 → $\int$ → 계산량 ↑

7.2 VAE와 인코더

EM 알고리즘은!

$$\log p_\theta(x) = \int q(z) \log \frac{p_\theta(x,z)}{q(z)} dz + \underbrace{D_{KL}(q(z) \| p_\theta(z|x))}_{\text{KL발산}}$$

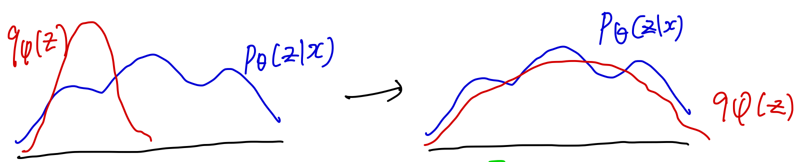
$$\geq \underbrace{\int q(z) \log \frac{p_\theta(x,z)}{q(z)}}_{\text{ELBO}}$$

EM 알고리즘
- GMM
  - $q(z)$ 갱신 → $q(z) = p_\theta(z|x)$ (사후 분포 계산)
  - θ 갱신 → ELBo 최대화
- VAE
  - 사후 분포 계산 어려움 → $q(z)$를 간단한 확률분포로 제한! (정규분포)
  - 한정된 확률 분포 내에서 ELBo 최대화

즉! $q_\psi(z) = N(z; \mu, \Sigma)$ → 한정!

따라서 → $$\log p_\theta(x) = \underbrace{\int q_\psi(z) \log \frac{p_\theta(x,z)}{q_\psi(z)}}_{\text{ELBO!}} dz + D_{KL}(q_\psi(z) \| p_\theta(z|x))$$

ELBo 최대화 → $$\text{ELBO}(x; \theta, \psi) = \int q_\psi(z) \log \frac{p_\theta(x,z)}{q_\psi(z)} dz$$



즉, $p_\theta(z|x)$와 $q_\psi(z)$가 같지 않아!

즉! 근사!를 해야해!

이걸! 변분 근사 (variational approximation) or 변분 베이즈 (variational bayes) 라고 해!

그럼 전체 데이터에서는 ?!?!

$$\sum_{n=1}^{N} ELBO(x^{(n)}; \phi, \psi^{(n)}) = \sum_{n=1}^{N} \int q_{\psi^{(n)}}(z) \log \frac{p_\theta(x^{(n)}, z)}{q_{\psi^{(n)}}(z)} dz$$

$x^{(n)}$은 .. N개 ...

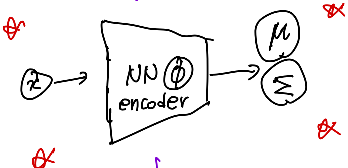
→ 근데 이걸 $q_{\psi}^{(n)}(z)$가 N개 필요..

너무 많아!

↓

그래서! 신경망!으로 대체!

↓



↓

즉! 근사 사후 분포의 매개변수(ψ)는 신경망으로 계산하는 방법은!!

분할 상환 추론 !

(amortized inference)

즉!

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_H \end{pmatrix}, \quad \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_H \end{pmatrix}, \quad \sigma = \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_H \end{pmatrix}, \quad z = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_H \end{pmatrix}$$

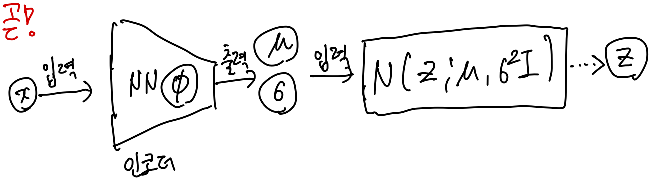
따라서! VAE 인코더가 하는 일은!

$$\mu, \sigma = NeuralNet(x; \phi)$$

Σ → 공분산 행렬

$$q_\phi(z|x) = N(z; \mu, \sigma^2 I)$$

곧!



7.3 ELBO 최적화

VAE 디코더

$$p(z) = N(z; 0, I)$$

$$\hat{x} = \text{NeuralNet}(z; \theta)$$

$$p_\theta(x|z) = N(x; \hat{x}, I)$$

VAE 인코더

$$\mu, \sigma = \text{NeuralNet}(x; \phi)$$

$$q_\phi(z|x) = N(z; x, \mu, \sigma^2 I)$$

하나의 데이터에 대한 ELBO

$$\text{ELBO}(x; \theta, \phi) = \int q_\phi(z|x) \log \frac{p_\theta(x, z)}{q_\phi(z|x)} dz$$

여러 개의 데이터에 대한 ELBO

$$\sum_{n=1}^{N} \text{ELBO}(x^{(n)}; \theta, \phi) = \sum_{n=1}^{N} \int q_\phi(z|x^{(n)}) \log \frac{p_\theta(x^{(n)}, z)}{q_\phi(z|x^{(n)})} dz$$

2번 ELBO 최대화 공식은 ?!?!

$$\text{ELBO}(x; \theta, \phi) = \int q_\phi(z|x) \log \frac{p_\theta(x, z)}{q_\phi(z|x)} dz$$

$$= \int q_\phi(z|x) \log \frac{p_\theta(x|z)\, p(z)}{q_\phi(z|x)} dz$$

$$= \int q_\phi(z|x) \log p_\theta(x|z)\, dz - \int q_\phi(z|x) \log \frac{q_\phi(z|x)}{p(z)} dz$$

$$= \underbrace{E_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right]}_{J_1} - \underbrace{D_{KL}\left(q_\phi(z|x) \,\|\, p(z)\right)}_{J_2}$$

J_1 = 기댓값
= 재구성 오차
(reconstruction error)
or
재구성 오류

J_2 = KL 발산
= 정규화
(regularization)
or
일관성 항

J_1 계산 → 몬테카를로!

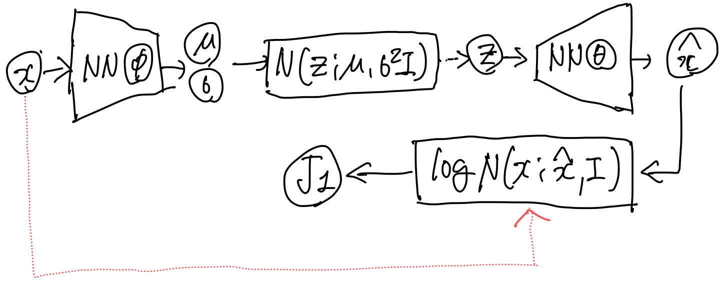
즉! $\mu, \sigma = \text{NeuralNet}(x; \phi)$

$z \sim N(z; \mu, \sigma^2 I)$

$\hat{x} = \text{NeuralNet}(z; \theta)$

$J_1 \approx \log N(x; \hat{x}, I)$

⇒



$$J_1 \approx \log N(x; \hat{x}, I)$$

$$= \log\left(\frac{1}{\sqrt{(2\pi)^D |I|}} \exp\left(-\frac{1}{2}(x-\hat{x})^T I^{-1} (x-\hat{x})\right)\right)$$

$$= -\frac{1}{2}(x-\hat{x})^T(x-\hat{x}) + \underbrace{\log\frac{1}{\sqrt{(2\pi)^D}}}_{\text{const: 상수}}$$

$$= -\frac{1}{2}\sum_{d=1}^{D}(x_d-\hat{x}_d)^2 + \text{const}$$

J_2 계산

$q_\phi(z|x) = N(z; \mu, \sigma^2 I)$

$p(z) = N(z; 0, I)$

⇒

$$J_2 = D_{KL}(q_\phi(z|x) \,\|\, p(z))$$

$$= -\frac{1}{2}\sum_{h=1}^{H}(1 + \log\sigma_h^2 - \mu_h^2 - \sigma_h^2)$$

즉! ELBO는!

$$\text{ELBO}(x; \theta, \phi) \approx -\frac{1}{2}\sum_{d=1}^{D}(x_d-\hat{x}_d)^2 + \frac{1}{2}\sum_{h=1}^{H}(1 + \log\sigma_h^2 - \mu_h^2 - \sigma_h^2) + \text{const}$$

드디어 ㅠㅠ

이때! $z \sim N(x; \mu, \sigma^2 I)$에서 z에서 미분이 곤란!

따라서, 재매개변수화 트릭 사용! ⇒ $\epsilon \sim N(\epsilon; 0, I)$ ⇒ $z = \mu + \sigma \odot \epsilon$

(reparameterization trick)

아다마르 곱

(Hadamard product)

7.4 VAE 구현

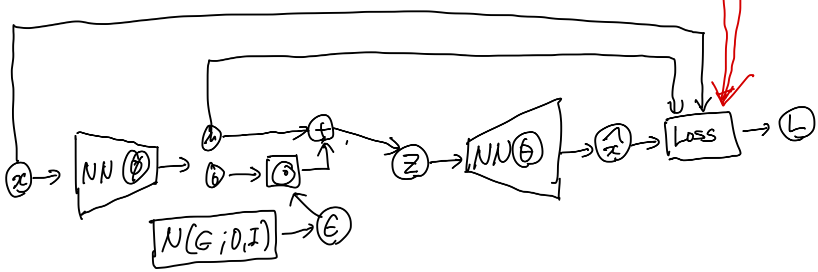
VAE 목표 = ELBO 최대화

$$ELBO(x;\theta,\phi) \approx -\frac{1}{2}\sum_{d=1}^{D}(x_d-\hat{x}_d)^2 + \frac{1}{2}\sum_{h=1}^{H}(1+\log\sigma_h^2-\mu_h^2-\sigma_h^2) + const$$

이는, 즉, 손실 함수 최소화, 따라서,

$$Loss(x;\theta,\phi) \approx \sum_{d=1}^{D}(x_d-\hat{x}_d)^2 - \sum_{h=1}^{H}(1+\log\sigma_h^2-\mu_h^2-\sigma_h^2)$$

즉, VAE의 학습 계산 그래프는,



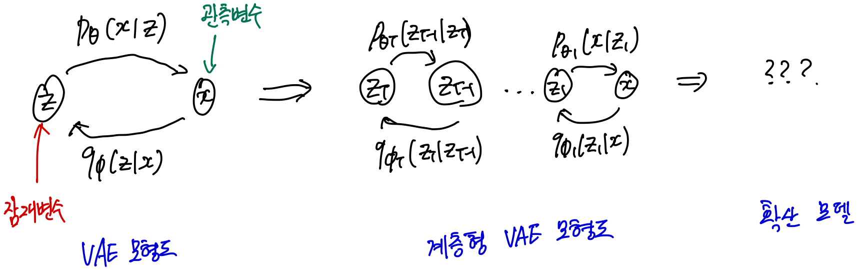
Ch 08. 확산 모델 이론

1. VAE에서 확산 모델로
2. 확산 과정에서 역확산 과정
3. ELBO 계산 ①
4. ELBO 계산 ②
5. ELBO 계산 ③
6. 확산 모델의 학습 (알고리즘)

8.1 VAE에서 확산 모델로

DDPM? → 확산모델

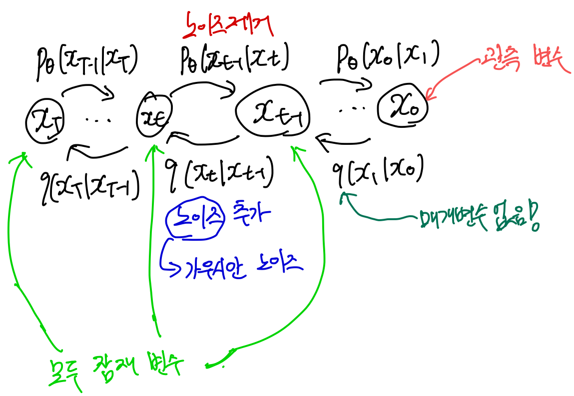
(Denoising Diffusion Probabilistic Models, 노이즈 제거 확산 확률 모델)



확산모델은?

계층형 VAE + ① 관측 변수와 잠재 변수의 차원 일치? ② 고정된 정규 분포를 따르는 노이즈를 인코더에 추가? ⇒ 확산모델?

즉? 확산모델 모형도? ⇒



8.2 확산 과정과 역확산 과정

확산 모델 — 확산 과정 : 노이즈를 추가하는 과정
　　　　　 — 역확산 과정 : 노이즈를 제거하는 과정

확산 과정(diffusion process)

완전한 노이즈

$q(x_t|x_{t-1})$

미리 설정한 값!

즉! $q(x_t|x_{t-1}) = N(x_t; \sqrt{1-\beta_t}\, x_{t-1}, \beta_t I)$

그럼! $p(x_T) \approx N(x_T; 0, I)$!

이때! 재매개변수화 트릭 발동!

$$\epsilon \sim N(\epsilon; 0, I)$$

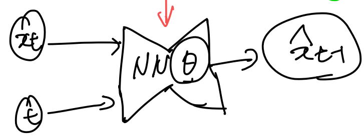
$$\therefore x_t = \sqrt{1-\beta_t}\, x_{t-1} + \sqrt{\beta_t}\, \epsilon$$

역확산 과정(reverse diffusion process)

$p_\theta(x_{t-1}|x_t)$

요건 추정되!

이는!



즉! $\hat{x}_{t-1} = \mathrm{NeuralNet}(x_t, t; \theta)$

$$\therefore p_\theta(x_{t-1}|x_t) = N(x_{t-1}; \hat{x}_{t-1}, I)$$

요건 그냥 예시(행렬)

정확히는! 공분산행렬!

8.3 ELBO 계산 ①

확산 모델 $\longrightarrow$ ELBO $\longrightarrow$ 근삿값 $\xrightarrow{\text{우선!}}$ 최종 시각 $=T$ $\Rightarrow$ T개 샘플 데이터 $x_1, x_2 \cdots x_T$ 근삿값!

3가지 변경

1. $x \to x_0$
2. $z \to x_{1:T}$
3. ϕ 삭제

확산 모델의 ELBO

$$\text{VAE의 ELBO}(x;\theta,\phi) = \int q_\phi(z|x) \log \frac{p_\theta(x,z)}{q_\phi(z|x)} dz$$

$$= E_{q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right]$$

$$\Rightarrow \text{확산 모델의 ELBO}(x_0;\theta) = E_{q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right]$$

알아두라!

$$p_\theta(x_{0:T}) = p_\theta(x_0|x_1)\,p(x_1|x_2)\cdots p_\theta(x_{T-1}|x_T)\,p(x_T)$$

$$= p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$$

$\xrightarrow{\text{즉!}}$

$$\text{ELBO}(x_0;\theta)$$

$$= E_{q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right]$$

$$= E_{q(x_{1:T}|x_0)}\left[\log \frac{p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)}{\prod_{t=1}^{T} q(x_t|x_{t-1})}\right]$$

$$= E_{q(x_{1:T}|x_0)}\left[\underbrace{\log \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)}_{\theta\text{ 포함!}} + \underbrace{\log \frac{p(x_T)}{\prod_{t=1}^{T} q(x_t|x_{t-1})}}_{\theta\text{ 미포함}}\right]$$

즉!

$$J(\theta) = E_{q(x_{1:T}|x_0)}\left[\log \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)\right]$$

$$= E_{q(x_{1:T}|x_0)}\left[\sum_{t=1}^{T} \log p_\theta(x_{t-1}|x_t)\right]$$

따라서! 몬테 카를로 사용!

⇓

여러 $x_{1:T}$ 생성

⇓

$\sum_{t=1}^{T} \log p_\theta(x_{t-1}|x_t)$ 평균 계산

$\xrightarrow{\text{샘플크기가 1이라면}}$

$$x_{1:T} \sim q(x_{1:T}|x_0)$$

$$J(\theta) \approx \sum_{t=1}^{T} \log p_\theta(x_{t-1}|x_t)$$

즉!

원본 $x_0 \xrightarrow{\text{생성}} x_{1:T} \xrightarrow{\text{각 시각에서 계산}} \log p_\theta(x_{t-1}|x_t)$

$$\hat{x}_{t-1} = \text{NeuralNet}(x_t, t; \theta)$$

$$p_\theta(x_{t-1}|x_t) = N(x_{t-1}; \hat{x}_{t-1}, I)$$

또한!

$$J(\theta) \approx \sum_{t=1}^{T} \log p_\theta(x_{t-1}|x_t)$$

$$= \sum_{t=1}^{T} \log N(x_{t-1}; \hat{x}_{t-1}, I)$$

t 시각 번호 변경.

정규 분포 식 대입

$$= \sum_{t=0}^{T-1} \log \frac{1}{\sqrt{(2\pi)^D |I|}} \exp\left\{-\frac{1}{2}(x_t - \hat{x}_t)^T I^{-1} (x_t - \hat{x}_t)\right\}$$

$$= -\frac{1}{2} \sum_{t=0}^{T-1} (x_t - \hat{x}_t)^T (x_t - \hat{x}_t) + \underbrace{T \log \frac{1}{\sqrt{(2\pi)^D}}}_{\text{상수} \rightarrow \text{무시!}}$$

$$= -\frac{1}{2} \sum_{t=0}^{T-1} \|x_t - \hat{x}_t\|^2$$

즉! 목적 함수 $J(\theta)$는!

1. 확산 과정을 통해 T개 샘플 얻기!
2. 신경망을 T번 적용해서 노이즈 제거!
3. 각 시각의 제곱오차 $\|x_t - \hat{x}_t\|^2$ 계산!

⇒ 'T'번이나...? 너무 많잖아...

8.4 ELBO 계산 ②

역시! T개는 너무 많아! $\xrightarrow{\text{따라서}}$ 2개로 해보자!

즉! $q(x_t|x_0) = N(x_t; \sqrt{\bar{\alpha}_t}\,x_0, (1-\bar{\alpha}_t)I)$ $\xrightarrow{\text{기호!}}$ $\alpha_t = 1-\beta_t$

$\bar{\alpha}_t = \alpha_t\alpha_{t-1}\cdots\alpha_1$

우선 $J(\theta)$을 확장!

$$J(\theta) = E_{q(x_{1:T}|x_0)}\left[\sum_{t=1}^{T} \log p_\theta(x_{t-1}|x_t)\right]$$

기댓값의 선형성

$$= \sum_{t=1}^{T} E_{q(x_{1:T}|x_0)}\left[\log p_\theta(x_{t-1}|x_t)\right]$$

관련 변수의 기댓값

$$= \sum_{t=1}^{T} E_{q(x_{t-1},x_t|x_0)}\left[\log p_\theta(x_{t-1}|x_t)\right]$$

즉! ⟹ 2번의 샘플링만 해도 돼요!

요기에! 관련X 따라서! 가능 O

그리고!

이것도 너무 커! 즉! 균등분포 + 몬테카를로! 즉! $J(\theta) = \sum_{t=1}^{T} E_{q(x_{t-1},x_t|x_0)}\left[\log p_\theta(x_{t-1}|x_t)\right]$

$$= T\,E_{u(t)}\left[E_{q(x_{t-1},x_t|x_0)}\left[\log p_\theta(x_{t-1}|x_t)\right]\right]$$

⇓ 몬테카를로!

요게!

$$E_{u(t)}[f(t)] = \sum_{t=1}^{T} u(t)f(t) = \sum_{t=1}^{T}\frac{1}{T}f(t) = \frac{1}{T}\sum_{t=1}^{T}f(t)$$

$$\therefore T\,E_{u(t)}[f(t)] = \sum_{t=1}^{T}f(t)$$

cf 샘플 = 1개

$t \sim U\{1,T\}$

$x_{t-1} \sim q(x_{t-1}|x_0)$

$x_t \sim q(x_t|x_{t-1})$

$\Longrightarrow J(\theta) \approx T\log p_\theta(x_{t-1}|x_t)$ 즉!

$\hat{x}_{t-1} = \text{NeuralNet}(x_t, t; \theta)$

$p_\theta(x_{t-1}|x_t) = N(x_{t-1}; \hat{x}_{t-1}, I)$

⇓ $J(\theta)$는?!?!

$$J(\theta) \approx -\frac{T}{2}\|x_{t-1} - \hat{x}_{t-1}\|^2$$

⟸ $J(\theta)$ 계산은?

1. 균등분포 $U\{1,T\}$에서 시각 t를 샘플링
2. $q(x_{t-1}|x_0)$에서 x_{t-1}을 샘플링
3. $q(x_t|x_{t-1})$에서 x_t를 샘플링
4. 신경망에 x_t를 입력! $\hat{x}_{t-1}$를 출력!
5. 제곱오차 $\|x_{t-1} - \hat{x}_{t-1}\|^2$ 계산!

8.5 ELBO 계산 ③

2번! 이제는 단 하나의 샘플 데이터만 사용해서 ELBO을 근사!

핵심은! 정규 분포!

$q(x_{t-1}|x_t, x_0)$ 뜻: x_0와 x_t가 주어졌을 때 x_{t-1}의 확률! $\xrightarrow{\text{중요!}}$ 해석적 풀이가 가능

$$x_t \xleftarrow{q(x_t|x_{t-1})} x_{t-1} \leftarrow \cdots \leftarrow x_1 \xleftarrow{q(x_1|x_0)} x_0$$

$q(x_{t-1}|x_t, x_0)$!

$$q(x_{t-1}|x_t, x_0) = N(x_{t-1}; \mu_q(x_t, x_0), \sigma_q^2(t) I)$$

$$\mu_q(x_t, x_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})x_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)x_0}{1-\bar{\alpha}_t}$$

$$\sigma_q^2(t) = \frac{\overbrace{(1-\alpha_t)}^{1-\beta_t}(1-\bar{\alpha}_{t-1})}{1-\underbrace{\bar{\alpha}_t}_{\alpha_t\alpha_{t-1}\cdots\alpha_1}}$$

중요점!

이전 두 개 ELBO $\xrightarrow{q(x_{t-1}|x_t, x_0)\text{ 형태!}}$ 한 개 ELBO

즉! $$J(\theta) = \sum_{t=1}^{T} E_{q(x_{t-1}, x_t|x_0)}[\log p_\theta(x_{t-1}|x_t)]$$

$$= T E_{u(t)}\Big[\underbrace{E_{q(x_{t-1}, x_t|x_0)}[\log p_\theta(x_{t-1}|x_t)]}_{J_0}\Big] \Rightarrow T E_{u(t)}[J_0]$$

여기에 $q(x_{t-1}|x_t, x_0)$를 추가!

따라서!

$$\arg\max_\theta J_0 = \arg\max_\theta \Big(J_0 - \overbrace{E_{q(x_{t-1}, x_t|x_0)}[\log q(x_{t-1}|x_t, x_0)]}^{\text{그냥 상수!}}\Big)$$

$$= \arg\max_\theta E_{q(x_{t-1}, x_t|x_0)}[\log p_\theta(x_{t-1}|x_t) - \log q(x_{t-1}|x_t, x_0)]$$

$$= \arg\max_\theta \underbrace{E_{q(x_{t-1}, x_t|x_0)}\left[\log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)}\right]}_{J_1} \xrightarrow{\text{새로운 } J(\theta)} J(\theta) = T E_{u(t)}[J_1]$$

New!

그럼 J_1은? 어떤 식 전개?

$$J_1 = \int q(x_{t-1}, x_t | x_0) \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)} dx_{t-1}\, dx_t$$

$q(x_{t-1}, x_t | x_0) = q(x_t|x_0)\, q(x_{t-1}| x_t, x_0)$

$$= -\int q(x_t|x_0) \underbrace{\int q(x_{t-1}|x_t, x_0) \log \frac{q(x_{t-1}|x_t, x_0)}{p_\theta(x_{t-1}|x_t)} dx_{t-1}}_{\text{KL 발산}}\, dx_t$$

$$= -E_{q(x_t|x_0)}\left[D_{KL}\left(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t)\right)\right]$$

$\Longrightarrow$ 즉, $q(x_{t-1}|x_t, x_0) = p_\theta(x_{t-1}|x_t) \Longrightarrow$ 최대!

정규분포!

$$D_{KL}(\ldots) \Rightarrow \frac{1}{2\sigma_q^2(t)} \| \mu_\theta(x_t, t) - \mu_q(x_t, x_0)\|^2$$

즉!

신경망을 통해

$$\hat{x}_{t-1} = \text{NeuralNet}(x_t, t; \theta)$$

$$p_\theta(x_{t-1}|x_t) = N(x_{t-1}; \hat{x}_{t-1}, \sigma_q^2(t) I)$$

$\sigma_q^2(t)$: $q(x_{t-1}|x_t, x_0)$과 같은 값!

New!

$$J(\theta) = -T\, E_{u(t)}\left[E_{q(x_t|x_0)}\left[\frac{1}{2\sigma_q^2(t)} \| \mu_\theta(x_t, t) - \mu_q(x_t, t)\|^2\right]\right]$$

$$\text{Loss}(x_0; \theta) = E_{u(t)}\left[E_{q(x_t|x_0)}\left[\frac{1}{2\sigma_q^2(t)} \| \mu_\theta(x_t, t) - \mu_q(x_t, t)\|^2\right]\right]$$

$t \sim U\{1, T\}$

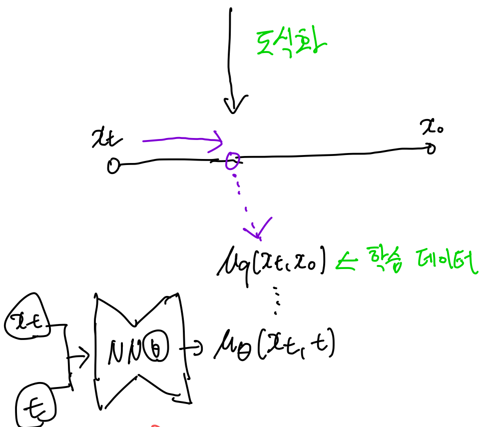
$x_t \sim q(x_t|x_0)$

$$= \frac{1}{\sigma_q^2(t)} \| \mu_\theta(x_t, t) - \mu_q(x_t, t)\|^2$$

8.6 확산 모델의 학습 (알고리즘)

확산 모델의 학습 알고리즘

1. 반복 :
2. x_0를 학습 데이터에서 무작위로 가져옴
3. $t \sim U\{1, T\}$, $\varepsilon \sim N(0, I)$
4. $x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \varepsilon$ ← $q(x_t|x_0)$에서 샘플링
5. $\mu_q(x_t, x_0) = \dfrac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})x_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)x_0}{1-\bar{\alpha}_t}$
6. $\sigma_q^2(t) = \dfrac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}$
7. $Loss(x_0 ; \theta) = \dfrac{1}{\sigma_q^2(t)} \left\| \mu_\theta(x_t, t) - \mu_q(x_t, x_0) \right\|^2$
8. $\dfrac{\partial}{\partial \theta} Loss(x_0 ; \theta)$ 갱신



$\mu_q(x_t, x_0)$를 학습 데이터로 하는 확산 모델!

원본 데이터를 복원하는 신경망

$$\mu_q(x_t, x_0) = \frac{\sqrt{\alpha_t}(1-\bar\alpha_{t-1})x_t + \bar\alpha_{t-1}(1-\alpha_t)x_0}{1-\bar\alpha_t}$$

대응!

$$\mu_\theta(x_t, t) = \frac{\sqrt{\alpha_t}(1-\bar\alpha_{t-1})x_t + \bar\alpha_{t-1}(1-\alpha_t)\boxed{\hat{x}_\theta(x_t, t)}}{1-\bar\alpha_t}$$

신경망이 추정한 값

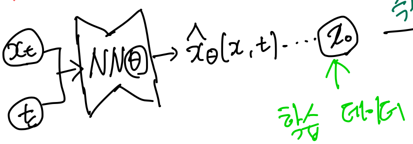
즉! $\Rightarrow$

$$D_{KL}(q(x_{t-1}|x_t, x_0) \,\|\, p_\theta(x_{t-1}|x_t))$$

$$= \frac{1}{2\sigma_q^2(t)} \|\mu_\theta(x_t, t) - \mu_q(x_t, x_0)\|^2$$

$$= \frac{1}{2\sigma_q^2(t)} \left(\frac{\sqrt{\bar\alpha_{t-1}}(1-\alpha_t)^2}{1-\bar\alpha_t}\right) \underbrace{\|\hat{x}_\theta(x_t, t) - x_0\|^2}_{\text{손실 함수로!}}$$

따라서!



즉! $\rightarrow$ x_0를 학습데이터로 하는 확산 모델!

노이즈를 예측하는 신경망

$$q(x_t|x_0) = N(x_t; \sqrt{\bar\alpha_t}x_0, (1-\bar\alpha_t)I)$$

$\Downarrow$ 재매개

$$\epsilon \sim N(0, I) \Longrightarrow x_t = \sqrt{\bar\alpha_t}x_0 + \sqrt{1-\bar\alpha_t}\,\epsilon \xrightarrow{\text{따라서}} x_0 = \frac{x_t - \sqrt{1-\bar\alpha_t}\,\epsilon}{\sqrt{\bar\alpha_t}}$$

즉!

$$\mu_q(x_t, x_0) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar\alpha_t}}\epsilon\right)$$

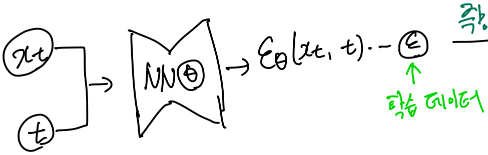
$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar\alpha_t}}\epsilon_\theta(x_t, t)\right)$$

즉! $\Rightarrow$

$$D_{KL}(q(x_{t-1}|x_t, x_0) \,\|\, p_\theta(x_{t-1}|x_t))$$

$$= \frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2}{(1-\bar\alpha_t)\alpha_t} \underbrace{\|\epsilon_\theta(x_t, t) - \epsilon\|^2}_{\text{손실 함수}}$$

따라서!



즉! $\rightarrow$ ϵ를 학습 데이터로 하는 확산 모델!

새로운 데이터 샘플링

$$p_\theta(x_{t-1}|x_t) = N(x_{t-1}; \mu_\theta(x_t, t), \sigma_q^2(t) I)$$

$\downarrow$ 재매개화

$$\epsilon \sim N(0, I) \xrightarrow{\text{샘플링}} x_{t-1} = \mu_\theta(x_t, t) + \sigma_q(t)\epsilon$$

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right)$$

$$\sigma_q(t) = \sqrt{\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}}$$

확산 모델의 데이터 생성 방법

1. $x_T \sim N(0, I)$
2. for t in $[T \dots 1]$:
3. $\quad \epsilon \sim N(0, I)$
4. $\quad$ if $t=1$ then $\epsilon = 0$
5. $\quad \sigma_q(t) = \sqrt{\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}}$
6. $\quad x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sigma_q(t)\epsilon$
7. return x_0

Ch 10. 확산 모델 응용

1. 조건부 확산 모델
2. 점수 함수
3. 분류기 가이던스
4. 분류기 없는 가이던스
5. 스테이블 디퓨전

10.1 조건부 확산 모델

기존

$p(x)$ 모델링

조건부

$p(x|y)$ 모델링 (y: 조건)

↳ 텍스트, 이미지..

기존 신경망 모델링

$x_t,\ t \to \text{NN}_\theta \to \mu_\theta(x_t, t)$ ⇒ t-1 이미지를 예측하는 모델

$x_t,\ t \to \text{NN}_\theta \to \epsilon_\theta(x_t, t)$ ⇒ x_t에 추가된 노이즈를 예측하는 모델

$$p_\theta(x_{t-1}|x_t) = N(x_{t-1}; \mu_\theta(x_t, t), \sigma_q^2(t) I)$$

여기서! $p_\theta(x_0) = \int p_\theta(x_0, x_1, \dots, x_T)\, dx_1 \cdots dx_T$

$$= \int p_\theta(x_0|x_1)\, p_\theta(x_1|x_2) \cdots p_\theta(x_{T-1}|x_T)\, p(x_T)\, dx_1 \cdots dx_T$$

조건 추가!

$$p_\theta(x_0|y) = \int p_\theta(x_0|x_1, y) \cdots p_\theta(x_{T-1}|x_T, y)\, p(x_T)\, dx_1 \cdots dx_T$$

즉! $p_\theta(x_{t-1}|x_t, y) = N(x_{t-1}; \mu_\theta(x_t, t, y), \sigma_q^2(t) I)$

그럼 $\epsilon_\theta(x_t, t)$ 에서는?

$x_t,\ t,\ y \to \text{NN}_\theta \to \epsilon_\theta(x_t, t, y)$

y 추가 방법은?

$t \to$ 사인파 위치 인코딩 $\to o$

$y \to$ 임베딩 계층 $\to o$

$o + o \to v$ (벡터!)

10.2 점수 함수

$p(x|y)$ → 단순 조건 → 무시 가능성 존재

이를 위해: 가이던스 (guidance) 사용: $\xrightarrow{\text{이는:}}$ 점수 함수 (score function) 를 사용

↳ 안내, 지침, 지도

↳ 주어진 조건을 더 강조해 반영하는 구조

점수 함수는:

$$\epsilon \approx -\sqrt{1-\bar{\alpha}_t}\,\boxed{\nabla_{x_t} \log p(x_t)}$$

→ 로그 가능도 $\log p(x_t)$의 입력 x_t에 대한 기울기

즉: 점수 or 점수 함수:

즉:

$x_t, t \rightarrow NN_\theta \rightarrow \epsilon_\theta(x_t, t) \cdots \epsilon$ ⇒ 노이즈 ϵ을 추정하는 신경망

(가능도 기반 생성 모델, likelihood-based generative model)

$x_t, t \rightarrow NN_\theta \rightarrow s_\theta(x_t, t) \cdots -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\epsilon \approx \nabla_{x_t} \log p(x_t)$

점수 함수:에서 주목:

⇒ ϵ은 $\nabla_{x_t} \log p(x_t)$와 마이너스 상수배 $(-\sqrt{1-\bar{\alpha}_t})$ 만큼 다르다!!!!

⇓

점수를 추정하는 신경망

(점수 기반 생성 모델, score-based generative model)

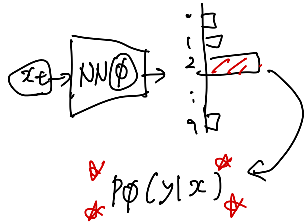
따라서:

$-\frac{1}{\sqrt{1-\bar{\alpha}_t}}\epsilon$을 학습 데이터로:

10.3 분류기 가이던스

분류기 가이던스 (classifier guidance) ⟹ 분류기 사용을 사용하여 데이터 생성을 안내하는

(classifier) ⇒ 데이터를 분류하는 학습된 신경망 ⇒



즉! x_t $\xrightarrow{\text{입력}}$ 신경망 $\xrightarrow{\text{출력}}$ 클래스 y ⟹ 일반적인 확산모델 $\xrightarrow{\text{확장}}$ 조건부 확산모델

시각 t에서의 노이지 이미지

이를 위한 조건!

1. 확산 모델은 점수 $\nabla_{x_t}\log p(x_t)$를 예측하는 신경망을 구현할 수 있다!
2. 조건부 확산 모델은 조건부 확률 점수 $\nabla_{x_t}\log p(x_t|y)$를 예측하는 신경망은 구현할 수 있다!

→ 분류기 가이던스!

우선!

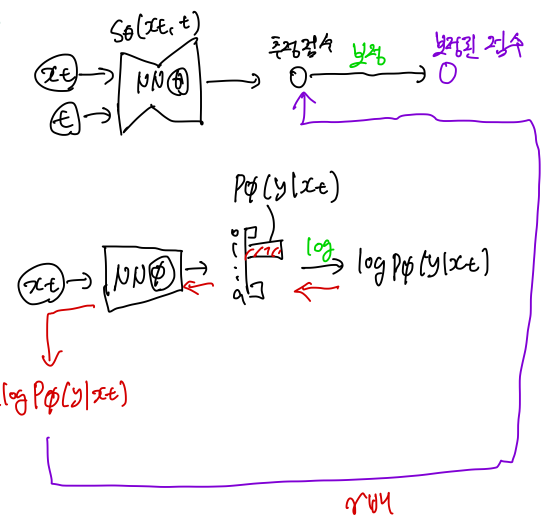
베이즈 정리

$$\underbrace{\nabla_{x_t}\log p(x_t|y)}_{\text{조건부 점수}} = \nabla_{x_t}\log\left(\frac{p(x_t)\,p(y|x_t)}{p(y)}\right)$$

$$= \underbrace{\nabla_{x_t}\log p(x_t)}_{\text{점수}} + \underbrace{\nabla_{x_t}\log p(y|x_t)}_{\text{분류기 로그 가능도의 기울기}} - \underbrace{\nabla_{x_t}\log p(y)}_{0}$$

점수 ↑계산 점수 예측 신경망 $s_\theta(x_t,t)$

분류기 로그 가능도의 기울기 ↑계산 분류기 신경망



즉! $\nabla_{x_t}\log p(x_t|y)$

$= \nabla_{x_t}\log p(x_t) + \gamma\nabla_{x_t}\log p(y|x_t)$ (γ: 분류기 기여도 가중치)

$\approx \underbrace{s_\theta(x_t,t)}_{\text{점수 추정 신경망}} + \gamma\underbrace{\nabla_{x_t}\log p_\phi(y|x_t)}_{\text{분류기}}$

시각화 ⟹

10.4 분류기 없는 가이던스 (classifier-free guidance)

분류기 없는 가이던스 공식?

$\nabla_{x_t} \log p(x_t \mid y)$ ---> 조건부 점수

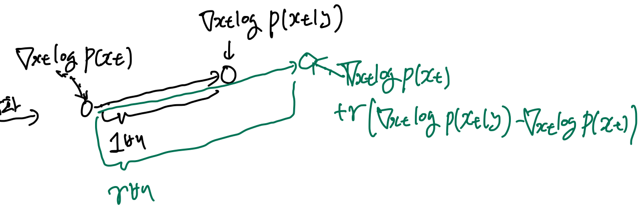
$= \nabla_{x_t} \log p(x_t) + \gamma \nabla_{x_t} \log p(y \mid x_t)$ ---> 분류기 가이던스

베이즈 정리

$$= \nabla_{x_t} \log p(x_t) + \gamma \Big(\underbrace{\nabla_{x_t} \log p(x_t \mid y)}_{\substack{\text{조건부 점수} \\ \uparrow \text{추정} \\ s_{\theta_1}(x_t, t)}} + \underbrace{\nabla_{x_t} \log p(y)}_{0} - \underbrace{\nabla_{x_t} \log p(x_t)}_{\substack{\text{조건 없는 점수} \\ \uparrow \text{추정} \\ s_{\theta_2}(x_t, t, y)}} \Big)$$

이는 곧?

$\nabla_{x_t} \log p(x_t)$ 에서 $\nabla_{x_t} \log p(x_t \mid y)$로 γ배 이동 $\xrightarrow{\text{시각화}}$



구현은?

하나의 신경망?으로 구현? → 조건 없는 점수 추정: $s_\theta(x_t, t, \phi)$
↳ 조건부 점수 추정: $s_\theta(x_t, t, y)$

즉? 분류기 없는 가이던스 공식은?

$$\nabla_{x_t} \log p(x_t \mid y) \approx s_\theta(x_t, t, \phi) + \gamma \big(s_\theta(x_t, t, y) - s_\theta(x_t, t, \phi) \big)$$

↓ 시각화