




Lecture 13 Background Knowledge

 Assign	
 Status	In Progress
 url	https://dnddnjs.github.io/paper/2018/06/19/vae/

VAE Tutorial 1: cs213n Lecture 13 - Generative Models

[#01. Supervised Learning & Unsupervised Learning](#)

[VAE의 중요 포인트 네 가지](#)

[#02. Generative Model](#)

[Taxonomy of Generative Models](#)

[#03. Variational Auto-Encoder](#)

[posterior를 approximate하는 새로운 함수 정의](#)

[#4. ELBO\(Evidence Lower Bound\)](#)

내가 가장 좋아하는 당근마켓 product manager 응원님께서 작성해주신 소중한 글이다.
cs231n만으로 부족했던 내용들을 채워나가보도록 하자!

- VAE Tutorial
- 이름부터 너무 마음에 든다!
- 순서는 cs231n 강의 리뷰, VAE 논문 & 코드 리뷰, Sentence VAE, Music VAE이다.
 - VAE Tutorial 목차
 - Tutorial 1: cs231n 강의 리뷰
 - Tutorial 2: VAE 논문 및 코드리뷰
 - Tutorial 3: SentenceVAE
 - Tutorial 4: MusicVAE
- Comment

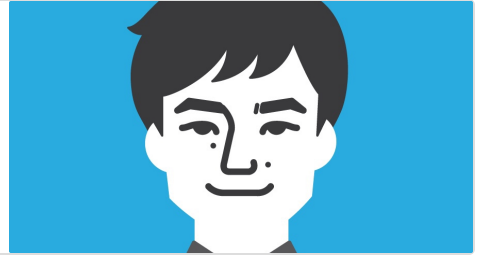
현업 딥러닝 엔지니어는 cs231n Lecture를 어떻게 공부했을까? 궁금해진다.
-20.06.16.tue-

Let's get it!

VAE Tutorial 1

cs231n 강의 내용과 Kingma의 논문을 통해 Variational Auto-Encoder를 정리해봅니다. 그리고 그 이후에 sequential data에 VAE를 적용한 사례인 SentenceVAE와 MusicVAE를 다룹니다. 다음과 같이 이

 <https://dnddnjs.github.io/paper/2018/06/19/vae/>



VAE Tutorial 1: cs213n Lecture 13 - Generative Models

#01. Supervised Learning & Unsupervised Learning

- Supervised Learning의 경우 학습을 위해서는 반드시(항상) label이 있어야함
- 이는 데이터에 대한 cost가 크다는 것을 의미
- 딥러닝의 경우 모델의 성능이 데이터의 양과 질에 크게 의존하기 때문에 비용적으로 부담이 큼
- 반면 Unsupervised Learning의 경우 input이 되는 x데이터만으로도 학습이 가능하다
- Unsupervised Learning을 통해 Data의 underlying hidden structure를 학습하는 것이 가능하다
- 대표적으로 Clustering, dimensionality reduction, feature learning, density estimation등이 가능하다.
- Unsupervised Learning에서는 K-means와 Auto-Encoder가 유명하다

VAE의 중요 포인트 네 가지

VAE를 제대로 이해하기 위해서는 코드부터 보는 것이 아니라 네 가지를 명심해야 한다.

1. VAE는 Generative Model이라는 점
2. Latent Variable이라는 것이 있으며 이를 바탕으로 데이터를 생성한다는 것(Decoder)
3. 문제를 더 쉽게 만들기 위해 Latent Variable이라는 것을 Encoder를 통해 추출한다는 것
4. VAE의 학습과정은 MLE라는 것

Woongwon님께서서는 위의 네 가지가 중요하다고 말씀해 주셨다.

#02. Generative Model

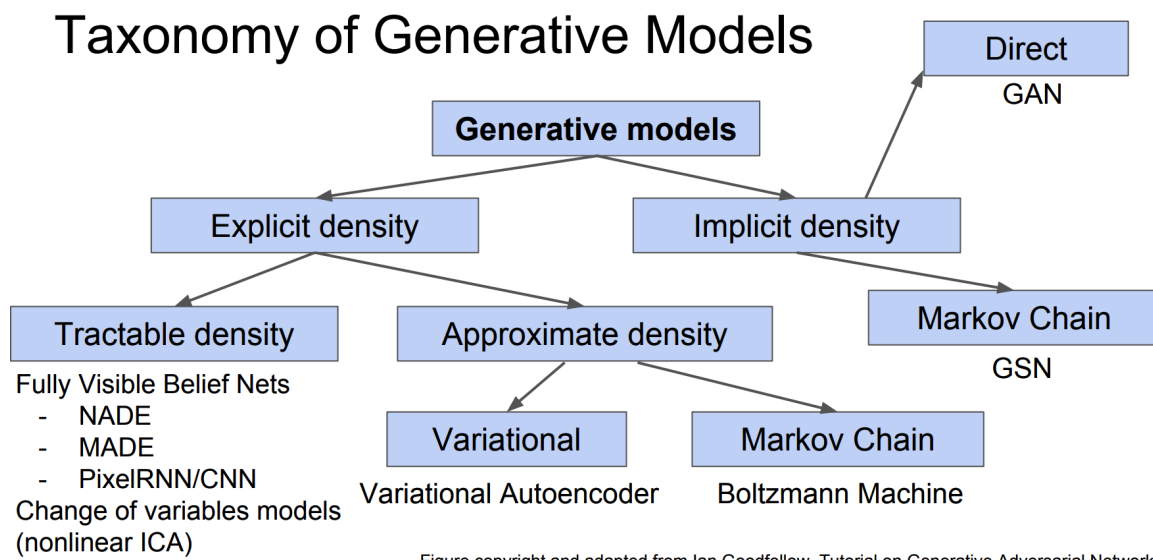
- VAE는 일종의 Generative Model이라고 봐야한다.
- Generative 모델이란 무엇인가?
 - training data가 주어졌을 때 이 data가 sampling된 분포와 같은 분포에서 새로운 sample을 생성하는 model이다.

- 즉 $P_{\text{model}}(X)$ 가 최대한 $P_{\text{data}}(X)$ 에 가깝게 만드는 것이 목표이다.
- 결국 얼마나 기존 모델과 가까운 것인가에 대한 지표를 만들어야하고 그 차이를 최소화하도록 gradient를 계산해서 업데이트 해야하는 것이다.

Q1. Explicit density estimation

Q2. Implicit density estimation

Taxonomy of Generative Models

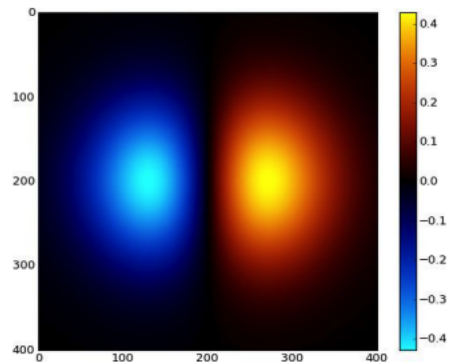
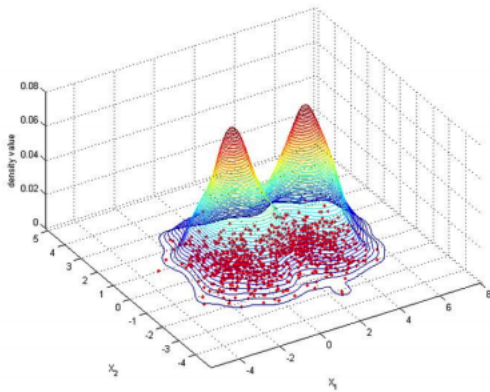


- 위의 그림은 GAN의 창시자 Ian Goodfellow가 정리한 도표이다.
- Generative Model은 크게 Explicit Density와 Implicit Density 두 가지로 나눌 수 있다.
- Explicit Density 모델은 data를 샘플링한 모델의 구조를 명확히 정의한다.
 - 정확히 정의한 모델로부터 data를 sampling하는 것이다.
- 반면 Implicit Density에서는 모델에 대한 구조를 explicit하게 정의하지 않는다.
 - 예를들어, GAN의 경우 noise로부터 data로의 transformation을 학습한다.
- VAE는 data를 sampling할 density model을 explicit하게 정의해서 직접적으로 학습하는 경우라고 할 수 있다.



Figure copyright Ian Goodfellow, 2016. Reproduced with permission.

1-d density estimation



2-d density estimation

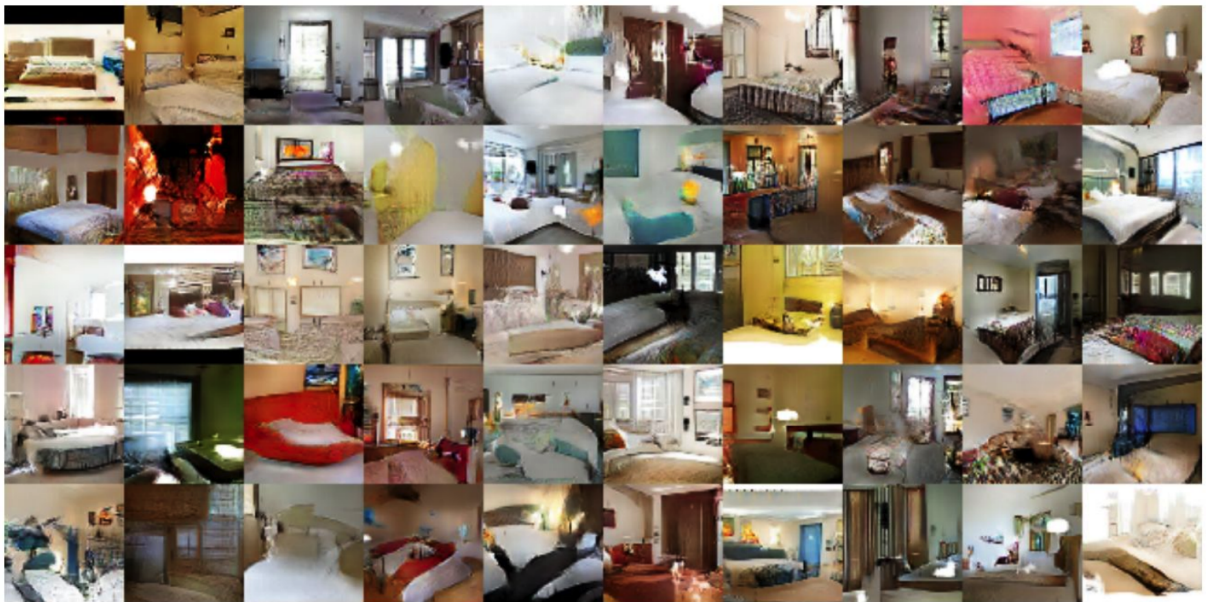
• 정리

- density estimation은 x 라는 데이터만 관찰할 수 있을 때, 관찰할 수 없는 x 가 샘플링된 확률 밀도 함수(probability density function)을 estimate하는 것이다.
- 다시 말해 Density estimation이 하는 것은 이 데이터 포인트가 많은 곳은 확률이 높고 데이터 포인트가 없는 곳은 확률이 낮아지는 것이다.
- 이러한 확률분포(파란색 선)가 있다면 현재 데이터 포인트와 같지는 않지만 비슷한 데이터를 생성해 낼 수 있는 것이다.

위의 그림과 같이 파란색선의 확률분포를 얻을 수 있다면 데이터 포인트와 같지는 않지만 비슷한 데이터를 생성해 낼 수 있다는 것이다.

- 즉 확률밀도함수(최대한 $P_{\text{data}}(X)$ 에 가까운 $P_{\text{model}}(X)$ 를 찾아냈을 때) $P_{\text{model}}(X)$ 를 통해 새로운 데이터를 generate할 수 있으므로 Generative Model이라고 부르는 것이다.
- 예를 들어 이미지의 경우 각 pixel은 0에서 255까지 나타낼 수 있으며 rgb는 3차원이므로 나타낼 수 있는 이미지의 양은 어마어마하게 많다.
- 하지만 우리가 realistic이라고 느끼는 이미지들은 그 중에 정말 일부이다. 또한 데이터셋에 있는 이미지들은 그보다 더 일부일 것이다.
- Generative Model은 그 엄청나게 큰 Space에서 realistic한 이미지를 샘플링할 수 있는 확률밀도함수이다.

Generative Model중에 가장 선명하고 가장 진짜같은 이미지를 생성하는 것은 GAN이다. GAN은 여러 변형체들이 많은데 그 중에 하나의 학습된 모델을 가지고 생성한 데이터 예시이다.



하지만 GAN의 경우 random noise로부터 데이터를 생성해내므로 데이터의 의미있는 representatrion을 학습할 수 없다. 이제 VAE에 대해 살펴보도록 한다.

#03. Variational Auto-Encoder

앞으로 MNIST 데이터셋을 생성하는 Generative Model에 대해서 이야기하도록 한다.

일단 probability density function을 정의한다.

$P_{\theta}(X)$ 라고 정의합니다. dataset은 $X = [X(i)]_{i=1, N}$ 라고 할 수 있습니다. 이 때 dataset의 datapoint($X(i)$)는 i.i.d하다고 가정합니다.

이 density function θ 라는 parameter가 정해졌을 때, X 라는 데이터가 나올 확률이다. 이 확률을 최대화하는 것이 Generative Model 혹은 density estimation의 목표입니다.

이 식은 z 라는 latent variable을 사용해서 다음과 같이 쓸 수 있습니다. 앞으로 우리는 이 식을 미분해서 그 미분값에 따라 stochastic gradient ascent를 할 것입니다.

$$p_{\theta}(x) = \int p_{\theta}(z)p_{\theta}(x|z)dz - (1)$$

VAE에서는 Auto-Encoder와 달리 latent variable을 정의한다.

그렇다면 왜 갑자기 latent variable라는게 등장했을까?

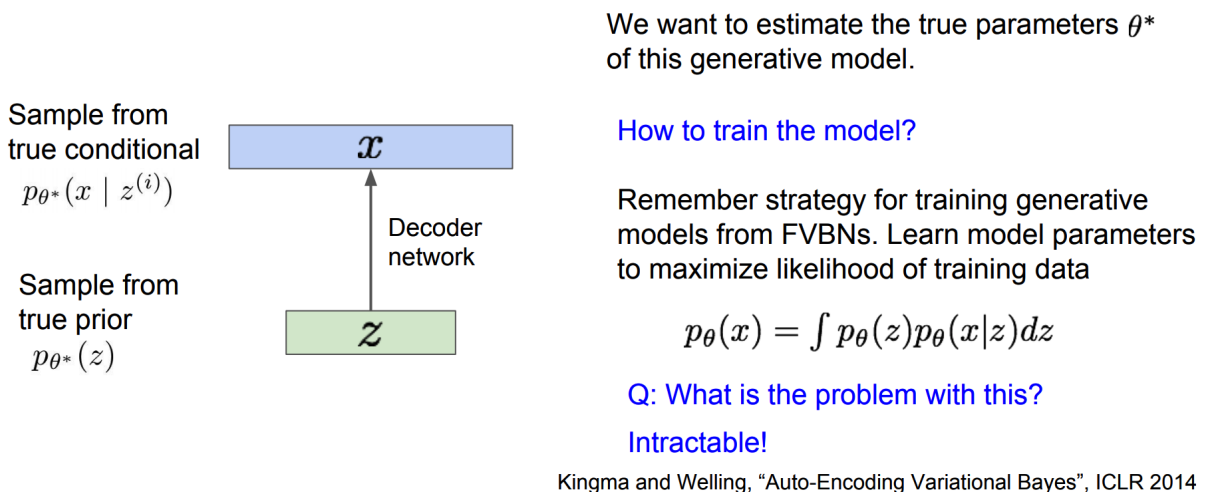
우리가 생성하고 싶은 데이터들은 상당히 차원이 높고 예를 들어 data point의 pixel 사이에 복잡한 관계가 있다. 따라서 pixel 사이의 관계를 확률모델로 모델링하는 것이 아니라 데이터를 표현하는 z 로부터 생성하는 모델을 만드는 것이다.

즉 데이터를 표현하는 latent vector z 로부터 데이터를 생성하는 graphical model을 생각해 보는 것이다.

(1)식의 우변에서 $p_{\theta}(z)$ 는 latent variable z 를 sampling할 수 있는 확률밀도 함수이다. 그리고 $p_{\theta}(x|z)$ 는 z 가 주어졌을 때 x 를 생성해내는 확률밀도 함수이다.

여기서 중요한 점은 이 $p_{\theta}(x|z)$ 가 θ 에 대해 미분가능해야한다는 것이다.

VAE는 z 를 구성하는 문제와 integral 문제를 해결해줍니다.



θ^* 는 dataset이 sampling된 true distribution의 parameter입니다. (이 분포는 parameterized 되었다고 가정합니다.)

simple한 Gaussian분포로부터 $z(i)$ 를 sampling합니다. decode network $p_{\theta^*}(x|z)$ 로부터 데이터를 생성합니다.

여기서 우리가 하고싶은 것은 θ^* 을 estimate 하는 것입니다.

z 는 사실 mnist에서 어떤 숫자인지만 나타내야하는 것이 아니라 **각 숫자마다도 다른 복잡한 특징들을 나타내야 합니다.** 이것을 단순한 normal distribution으로 만들고 decoder network의 layer들이 알아서 data를 생성해내는데 필요한 정보를 추출하도록 하는 것입니다.

즉 여러층의 layer들이 있다면 앞의 층들은 normal distribution을 latent value로 변환해주는 일을 하는 거이고 뒤의 층들은 이 latent value를 가지고 realistic한 digit을 sampling할 수 있는 확률밀도함수를 만들어내는 것입니다.

두 번째 문제인 integral의 경우 integral을 다 계산하지 않고 Monte-Carlo estimation을 통해 estimate할 것입니다. 여기서 Bayesian이 등장합니다.

대부분의 z 에 대해서는 $p_{\theta}(x|z)$ 는 거의 0의 값을 가질 것입니다. 따라서 sampling이 상당히 많이 필요합니다. 데이터셋이 클 경우에 이것은 너무 cost가 큼니다.

좀 더 efficient하게 이 sampling과정을 진행하려면 data에 dependent하게 z 를 sampling할 필요가 있습니다.

$$p_{\theta}(x) = \int p_{\theta}(z)p_{\theta}(x|z)dz \approx \frac{1}{N} \sum_{i=1}^N p_{\theta}(x|z^{(i)}) - (2)$$

따라서 $p_{\theta}(z|x)$ 를 생각해보는 것입니다. $p_{\theta}(z|x)$ 가 하는 역할은 x 가 주어졌을 때 이 x 를 생성해낼 것 같은 z 에 대한 확률분포를 만드는 것입니다.

Posterior density also intractable: $p_{\theta}(z|x) = p_{\theta}(x|z)p_{\theta}(z)/p_{\theta}(x)$

Intractable data likelihood

이것은 Bayes'rule에 의해 다음과 같이 쓸 수 있습니다.

하지만 식 (1)에서 보듯이 $p_{\theta}(x)$ 를 계산하는 것은 intractable하므로 이 posterior 또한 intractable posterior가 됩니다. 따라서 이 posterior를 approximate하는 새로운 함수를 정의합니다.

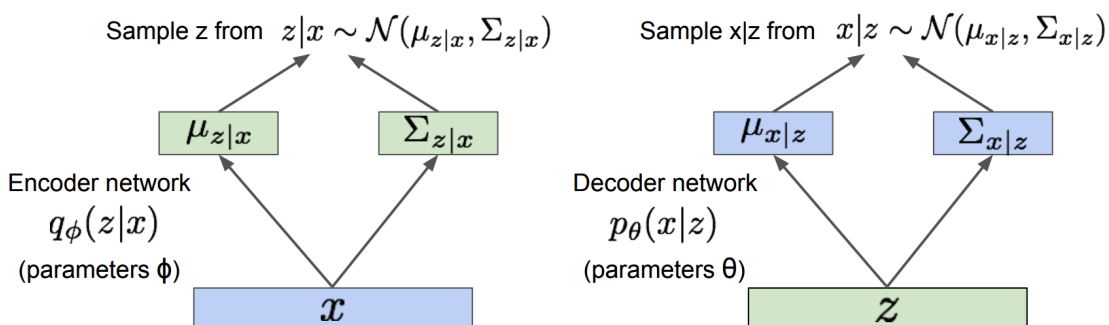
posterior를 approximate하는 새로운 함수 정의

ϕ 라는 새로운 parameter로 표현되는 $q_{\phi}(z|x)$ 는 일종의 encoder라고 볼 수 있다.

원래의 posterior를 approximate 했기 때문에 error가 존재한다. 따라서 원래의 objective function에 대한 lower bound를 정의할 것이다.

그 전에 VAE의 네트워크 구조를 살펴보자

- Encoder는 $q_{\phi}(z|x)$ 이며 x 를 input으로 받아서 z space상에서 확률분포를 만든다
- 이 확률분포는 gaussian이라고 가정해서 만든다.
- 이 data dependent한 gaussian 분포로부터 z 를 sampling한다.
- sampling된 z 를 가지고 decoder $p_{\theta}(x|z)$ 는 x 의 space 상의 gaussian distribution 혹은 Bernoulli distribution을 output으로 내놓는다.
- 그러면 x 를 이 분포로부터 sampling할 수 있다.
- 이러한 구조를 가지기 때문에 Auto-Encoder가 되는 것이며 학습이 되고 나면 latent variable z 라는 data의 의미있는 representation을 얻을 수 있다.



#4. ELBO(Evidence Lower Bound)

- 이제 VAE를 어떻게 학습시키는지 살펴보기 위해 objective function을 변형시켜보자
- log likelihood는 다음과 같다.
- 이 값을 최대화 시키는 것이 목표이다.

이제 VAE를 어떻게 학습시키는지 살펴보기 위해 objective function을 변형시켜보겠습니다. log likelihood는 다음과 같습니다. 이 값을 최대화시키는 것이 목표입니다. 이 식 자체는 intractable 하기 때문에 변형이 필요합니다.

$$\log p_{\theta}(x^{(i)})$$

이 log likelihood를 $q_{\phi}(z|x)$ 로부터 sampling한 latent variable z 에 대한 expectation 식으로 바꿀 수 있습니다. $p_{\theta}(x^{(i)})$ 가 z 에 dependent하지 않기 때문입니다.

$$\log p_{\theta}(x^{(i)}) = \mathbb{E}_{z \sim q_{\phi}(z|x^{(i)})}[\log p_{\theta}(x^{(i)})] - (3)$$

여기서 Bayes' Rule을 적용해보겠습니다.

$$p_{\theta}(z|x^{(i)}) = \frac{p_{\theta}(x^{(i)}|z)p_{\theta}(z)}{p_{\theta}(x^{(i)})}$$

$$p_{\theta}(x^{(i)}) = \frac{p_{\theta}(x^{(i)}|z)p_{\theta}(z)}{p_{\theta}(z|x^{(i)})} - (4)$$

(3)식에 (4)를 넣어서 다시 쓰면 다음과 같습니다.

$$\log p_{\theta}(x^{(i)}) = \mathbb{E}_{z \sim q_{\phi}(z|x^{(i)})}[\log \frac{p_{\theta}(x^{(i)}|z)p_{\theta}(z)}{p_{\theta}(z|x^{(i)})}] - (5)$$

그 다음에 expectation 안의 항에 같은 값을 곱하고 나눕니다.

$$\log p_{\theta}(x^{(i)}) = \mathbb{E}_{z \sim q_{\phi}(z|x^{(i)})}[\log \frac{p_{\theta}(x^{(i)}|z)p_{\theta}(z)}{p_{\theta}(z|x^{(i)})} \frac{q_{\phi}(z|x^{(i)})}{q_{\phi}(z|x^{(i)})}] - (6)$$

이 때, $p_{\theta}(z)$ 와 $q_{\phi}(z|x^{(i)})$ 를 하나로 묶고 $p_{\theta}(z|x^{(i)})$ 와 $q_{\phi}(z|x^{(i)})$ 를 하나로 묶어서 별도의 expectation으로 내보냅니다.

$$\log p_{\theta}(x^{(i)}) = \mathbb{E}_z[\log p_{\theta}(x^{(i)}|z)] - \mathbb{E}_z[\log \frac{q_{\phi}(z|x^{(i)})}{p_{\theta}(z)}] + \mathbb{E}_z[\log \frac{q_{\phi}(z|x^{(i)})}{p_{\theta}(z|x^{(i)})}] - (7)$$

(7)식에서 우변을 살펴보겠습니다. 우변의 두번째 항과 세번째 항은 잘 보면 KL-Divergence의 형태인 것을 알 수 있습니다. 따라서 KL의 형태로 바꿔쓰면 다음과 같습니다.

$$\log p_{\theta}(x^{(i)}) = \mathbb{E}_z[\log p_{\theta}(x^{(i)}|z)] - D_{KL}(q_{\phi}(z|x^{(i)})||p_{\theta}(z)) + D_{KL}(q_{\phi}(z|x^{(i)})||p_{\theta}(z|x^{(i)}))$$

