VAE

Auto-Encoding Variational Bayes

> Diederik P. Kingma, Max Welling

Abstract

다루기 힘든 사후 분포(intractable posterior distributions)을 가진 연속적인(continuous)한 잠재 변수(latent variable)가 존재하고 데이터셋이 큰 경우, 방향성 확률 모델에서 어떻게하면 효율적으로 추론(inference)와 학습(learning)을 수행할 수 있을지에 대한 당시 머신러닝의 중요하고 어려운 난제를 제시함.

> **방향성 확률 모델**(directed probabilistic models): 원인과 결과 관계 p(z) -> p(x|z)를 그래프로 나타내는 모델(directed graphical model)을 의미. 대표적으로 bayesian network가 있음.
> **잠재 변수**(latent variable): 데이터 x를 생성하지만 직접 관찰할 수는 없는 변수임. 예를 들어 이미지(x)를 생성하는 '얼굴 표정'이나 '글씨 스타일'(z)같은 특징임.
> **사후 분포**(posterior distribution): 관찰된 데이터 x가 주어졌을 때, 이 데이터를 생성했을 잠재 변수 z의 분포 p(z|x)를 의미함.

저자는 대규모 데이터셋에 확장 가능하며, 약간의 미분 가능성 조건 하에서는 다루기 힘든 경우에도 작동하는 확률적 변분 추론 및 학습 알고리즘을 소개함.

> **변분 추론**(variational inference): intractable한 사후 분포 $p(z|x)$를 $q(z|x)$라는 다루기 쉬운 근사 분포(예: 가우시안)로 대체하여 추론하는 기법
> **확률적**이라는 의미는 전체 데이터가 아닌 일부(미니 배치)를 무작위로 뽑아서 경사하강법으로 학습하는 것을 의미.(대용량 데이터셋에 적용 가능)

이 논문의 핵심 기여는 두가지가 있음.

첫번째로 변분 하한(variational lower bound, ELBO)의 재매개변수화(reparameterization)가 표준적인 확률적 경사 하강법(standard stochastic gradient methods)을 사용하여 간단하게 최적화될 수 있는 하한 추정치(lower bound estimator)를 산출한다는 것을 보여줌.

> **변분 하한**: $p(z|x)$를 $q(z|x)$로 근사할 때, 두 분포가 얼마나 다른지(KL-divergence)를 측정하는데, 이 과정에서 유도되는 목적 함수($\mathcal{L}(\theta, \phi; x^{(i)})$)임. VAE는 이 ELBO를 최대화하도록 학습함. 이때 ELBO는 $q_\phi(z|x)$에 대한 기댓값($\mathbb{E}_{q_\phi(z|x)}[\dots]$)을 포함함. 이 기댓값을 계산하기 위해 q_ϕ로부

> 터 z를 샘플링($z \sim q_\phi(z|x)$)해야 하는데, '샘플링' 연산은 미분이 불가능함. 따라서 q의 파라미터 ϕ로 gradient가 역전파될 수 없기에 ϕ를 학습시킬 수 없었음.
> **재매개변수화**: q에서 z를 샘플링($q_\phi(z|x)$) 대신 $z = g_\phi(\epsilon, x)$ (예: $z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon$, 여기서 $\epsilon \sim \mathcal{N}(0,1)$) 처럼 z를 x와 파라미터(ϕ)에 대한 결정론적 함수 g_ϕ와 외부 노이즈 ϵ의 결합으로 바꿈. 이로써 z를 샘플링하는 대신 ϵ을 샘플링하여 z를 ϕ에 대해 미분 가능한 함수로 표현할 수 있음. 따라서 전체 목적 함수를 ϕ에 대해 미분 가능하여 SGD로 최적화하는 것이 가능해짐.

두번째로 데이터포인트마다 연속적인 잠재 변수를 갖는 i.i.d. 데이터셋에 대해, 제안된 하한 추정치를 사용하여 다르기 힘든 사후 분포에 근사 추론 모델(encoder, 인식모델이라고도 함)을 피팅함으로써 사후 추론이 특히 효율적으로 수행될 수 있음을 보여줌.

> i.i.d.(independent and identically distributed)**: 데이터가 서로 독립적이고 동일한 분포에서 샘플링되었다는 표준적인 통계 가정임. 전통적인 변분 추론은 각 데이터포인트 x_i마다 최적의 z_i를 찾기 위해 별도의 최적화 과정이 필요했음.**
> 근사 추론 모델**: VAE는 $q_\phi(z|x)$를 신경망(Encoder)으로 만듦. 이 신경망은 x를 입력받아 최적의 q (예: q의 파라미터인 μ와 σ)를 한 번의 계산(forward pass)으로 출력함. 이로써 개별 데이터마다 z를 최적화하는 대신 x를 z의 분포 파라미터로 매핑하는 추론 모델 자체를 학습시킴.이 모델의 파라미터 ϕ는 모든 데이터에 대해 분담되어 공유됨. 따라서 학습이 끝난 후 새로운 x가 들어와도 즉시 z의 분포를 추론할 수 있어 효율적임.(인식모델은 x를 보고 z를 인식한다고 해서 붙여진 이름임.)

위 두가지 기여는 이후 실제 실험으로 보여짐.

> Q. intractable한 $p(z|x)$와 $q(z|x)$의 차이는 무엇인가? q(z|x)를 전개했을 때 q(x)를 포함하는 intractable한 분포가 아닌가?
> A. 실제 사후 분포 p(z|x)는 베이즈 정리에 의해 p(x|z)p(z)/p(x)로 표현되는데, p(x)(확률 변수 x에 대한 marginal likelihood)를 계산하는 것이 intractable함. $p(x) = \int p(x|z)p(z)dz$ 꼴의 적분이기 때문임. z는 연속 변수이고 p(x|z)는 복잡한 신경망(Decoder)이기 때문에, 모든 z에 대해 이 적분을 계산하는 것이 수학적으로 불가능함. 따라서 p(x)를 모르기 때문에 p(z|x)의 정확한 확률 밀도 값을 계산할 수 없음.

따라서 p(z|x)를 베이즈 정리를 통해 p(x)를 계산하는 것을 포기하고, 이를 근사하기 위해 $q_\phi(z|x)$를 함수(신경망)로 직접 정의함.

반면, q(z|x)는 우리가 선택한 근사 분포이므로 우리가 원하는 형태(예: 가우시안)로 정의할 수 있음. 따라서 q(z|x)는 다루기 쉬운 분포임.

1 Introduction

앞서 언급한 난제에 대해 사후분포를 정확하게 계산하는것이 불가능하기에 p(z|x)와 비슷한 q(z|x)를 효율적으로 찾는 것을 목표로 삼음.
변분 베이즈(Variational Bayes, VB) 접근 방식은 intractable한 사후 분포에 대한 근사치를 최적화하는 과정을 포함함. 하지만 일반적인 평균장(mean-field)접근 방식은 근사 사후 분포에 대한 기댓값의 해석적 해를 필요로 하는데, 이 역시 일반적인 경우에는 intractable함.
이 논문은 해석적 해를 포기하고 sampling을 통한 추정치(재매개변수화)를 사용함. 이 추정치가 편향되지 않다(unbiased)는 것은 추정치의 기댓값이 실제 하한값과 일치한다는 의미로 수학적으로 타당함을 보장함. 또, 이 추정치는 SGD로 최적화할 수 있을 만큼 간단해짐.

확률적 경사 변분 베이즈(Stochastic Gradient Variational Bayes, SGVB)추정치는 앞서 언급한 난제에 대해 효율적인 대안(재매개변수화, 근사 추론 모델)을 제공함.

> - SGVB: 재매개변수화 트릭을 사용하여 ELBO의 gradient를 계산하고 SGD로 최적화할 수 있게 만든 추정치.
> - AEVB: SGVB 추정치를 사용하여 인식 모델(encoder)을 도입하여, 데이터마다 추론을 반복하는 대신 추론 자체를 학습하는 알고리즘.
> - VAE: AEVB 알고리즘을 신경망으로 구현한 모델.

> Q. 해석적 해를 포기한다는 것은 어떤 의미인가?
> A. 먼저 해석적 해란 복잡한 방정식을 풀기 위해 논리적, 대수적 방법을 사용하여 정확한 형태의 해를 구하는 것을 의미. VB는 q(z)가 $p(z|x)$와 가장 비슷해지는 최적의 $q(z)$를 찾기 위해, 복잡한 기댓값 적분을 손으로 풀어야 했음. 하지만 이 적분이 복잡한 신경망일 경우에는 기댓값 적분이 복잡해져서 해석적으로 푸는 것이 불가능함. VAE는 이 해석적 해를 포기하고, Monte Carlo 샘플링을 사용하여 기댓값을 수치적으로 근사함으로써 이 문제를 해결함. 즉, 샘플 몇개 뽑아서 계산한 평균값으로 기댓값을 근사한다는 의미고 이게 SGVB임.
> $q(z) = \prod q(z_i)$
> Q. 평균장(mean-field)접근 방식이란 무엇인가?
> A. 평균장 이론(mean-field theory) 또는 자기 일관성 장 이론은 원래 모델의 자유도에 대해 평균화하여 근사하는 더 간단한 모델을 연구함을 의미함.[wikipedia] 이 개념은 본래 물리학(통계역학)에서 유래했지만, 변분 베이즈 추론에서 intractable한 사후 분포를 근사하기 위한 핵심 전략으로 사용됨. 다시 이 문제로 넘어와 생각하면 복잡한 확률 분포를 다루기 위해 복잡한 상호작용을 모두 무시하고, 단순한 독립적인 부분들의 곱으로 분해하여 근사하려는 아이디어임. 다시 말해, 근사하려는 분포(q(z))의 모든 잠재 변수가 서로 통계적으로 독립이라고 가정하는 것임. 수식으로 유도하면 $q(z) = \prod q(z_i)$ 이렇게 표현됨. 수학적으로 이는 $q(z)$가 z의 개별 변수 $z_1, z_2 \dots z_n$에 대한 분포의 곱으로 인수분해된다는 의미임. VAE 이전 전통적인 VB는 이 평균장 가정을 사용하여 해석적 해를 유도하려 했으나 복잡한 신경망에서는 불가능했음. 즉, 평균장 접근 방식은 q를 너무 단순하게 가정하는 것도 문제지만, 복잡한 모델(VAE의 decoder)에서는 해석적 해를 구하는 것 자체가 불가능하다는 문제가 있음.
> 3장에서 VAE의 $q_\phi(z|x)$를 대각 공분산(diagonal covariance)을 갖는 가우시안 $\mathcal{N}(\mu, \sigma^2 I)$으로 가정하는데 공분산 행렬이 대각이라는 것은 모든 잠재 변수(z_i, z의 모든 차원이)가 서로 독립이

> 라고 가정하는 것과 같음. VAE는 이 전략(전통적인 방식)을 지적하기보다는 SGVB라는 새로운 최적화 방식을 제시함.

> Q. 대각 공분산을 갖는 가우시안이란?
> A. 다변량 정규분포에서 잠재변수의 각 차원(z1, z2...)이 서로 독립이라고 가정하는 것. 수학적으로는 다변량 가우시안 분포의 공분산 행렬은 변수간의 상관관계를 나타내는데, 이때 공분산 행렬의 대각선 성분(분산, σ^2)을 제외한 나머지 모든 성분(공분산)이 0인 형태임. 이 과정을 통해 계산 복잡도를 크게 줄일 수 있음.

> Q. 재매개변수화 트릭에서 ϵ도 샘플링된 값이므로 미분이 불가능하지 않은가? 또, 샘플링 과정이 왜 미분할 수 없는가?
> A. 먼저 샘플링 과정이 왜 미분 불가능한지 다뤄보겠다. 미분(경사하강법)이 가능하려면, 파라미터(ϕ)의 작은 변화가 최종 손실(Loss)에 결정론적이고 연속적인 영향을 주어야 함. z는 $q_\phi(z|x)$에서 샘플링되는데 이 과정이 ϕ에 대해 결정론적이지 않고 확률적이기 때문에 ϕ에서 z로 이어지는 경로에 gradient가 흐를 수 없음. z는 ϕ에 대한 함수가 아닌 샘플일 뿐임.
> 그렇다면 ϵ도 샘플링되는데 어떻게 z에 대한 미분이 가능한가? reparameterization trick에서 z를 $z = g_\phi(\epsilon, x) = \mu_\phi(x) + \sigma_\phi(x)\epsilon$로 표현함. 이때 ϵ은 ϕ와 무관한 고정된 분포에서 샘플링.

ϕ는 ϵ을 샘플링하는 데 관여하는 것이 아니라, 샘플링된 ϵ을 z로 변환하는 결정론적 함수 g_ϕ에만 관여함. 이로써 최종손실에서 z까지의 gradient $\frac{\partial \mathcal{L}}{\partial z}$가 흐름.

> 정리하자면 ϵ라는 외부의 무작위성을 ϕ와 독립적으로 주입하고, ϕ는 이 무작위성을 z로 바꾸는 미분가능한 함수를 학습하게 만듦으로써 gradient가 흐를 수 있는 경로를 확보함.

i.i.d. 데이터셋과 연속적인 잠재 변수가 있는 경우를 위해 AEVB(auto-encoding variational bayes)라는 구체적인 SGVB 알고리즘을 제안함.
AEVB 알고리즘에서 SGVB 추정치를 사용하여 인식모델(encoder)을 최적화함으로써 inference와 learning을 효율적으로 만듦. 또한 데이터포인트마다 MCMC와 같은 반복적 추론 방식을 사용할 필요 없이 파라미터를 효율적으로 학습할 수 있음.
학습된 근사 사후 추론 모델은 인식(recognition), denoising, representation, visualization 등 다양한 작업에도 사용될 수 있음.

인식 모델에 신경망이 사용될 때, VAE에 도달함.

$$q_\phi(z|x) = EncoderNeuralNetwork$$

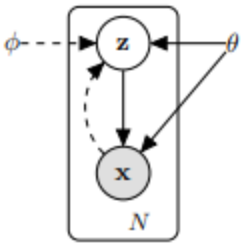
$$p_\theta(x|z) = DecoderNeuralNetwork$$

이처럼 q와 p가 신경망으로 구현하고, AEVB을 통해 학습시키는 전체 프레임워크를 변분 오토인코더(Variational Autoencoder, VAE)라고 부름.

> Q. 인코더와 디코더의 파라미터가 ϕ, θ로 다른 이유는?
> A. 서로 별개의 신경망(함수)이기 때문임.

2 Method

이 논문에서 제안하는 방법론이 VAE에만 국한되지 않고, 연속적인 잠재 변수를 가진 방향성 확률 모델 전반에 적용될 수 있음을 설명함.
데이터 포인트마다 잠재 변수가 있는 i.i.d. 데이터셋을 가지고 있고(각 데이터가 서로 독립적, 동일한 분포를 따름), (전역)파라미터(θ)에 대해서는 최대 우도 또는 최대 사후 확률(MAP) 추론을, 잠재 변수(z)에 대해서는 변분 추론을 수행하고자 하는 일반적인 경우로 한정함.



VAE가 다루는 확률모델의 구조를 도식화한 것.
실선은 생성모델(decoder) $p_\theta(z)p_\theta(x|z)$을 나타냄. z->x방향의 화살표는 z가 원인이 되어 데이터 x를 생성한다는 의미.
점선은 변분 근사(Variational approximation, encoder) $q_\phi(z|x)$를 나타냄. x->z방향의 화살표는 관찰된 데이터 x로부터 잠재 변수 z를 추론한다는 의미.
이 과정을 데이터셋의 크기 N만큼 독립적으로 반복됨.
θ는 생성모델의 파라미터, ϕ는 변분 모델의 파라미터이고, 이 둘은 함께 학습됨.

2.1 Problem Scenario

데이터가 서로 독립적(independent)이고 동일한 분포(identically distributed)를 따른다고 가정함.
데이터가 관찰되지 않은 연속 확률 변수 z를 포함하는 어떤 무작위 과정에 의해 생성된다고 가정할 때, 그 과정은 두 단계로 구성됨.

(1) 어떤 사전 분포(prior distribution) $p_{\theta^*}(z)$에서 값 $z^{(i)}$가 생성됨.
(2) 어떤 조건부 분포(conditional distribution) $p_{\theta^*}(x|z)$에서 값 $x^{(i)}$가 생성됨.
이때 θ^*는 알지 못하는 true 파라미터를 의미함.

우리는 사전 분포 $p_{\theta^*}(z)$와 우도(likelihood) $p_{\theta^*}(x|z)$가 파라미터 분포족 $p_\theta(z)$와 $p_\theta(x|z)$에서 왔다고 가정하고, 이들의 확률 밀도 함수(PDF)가 θ와 z 모두에 대해 거의 어디서나(almost everywhere) 미분 가능하다고 가정함.

> VAE가 작동하기 위한 두 가지 핵심 수학적 조건

1. 파라미터 분포 족(Parametric falilies): 우리가 찾으려는 확률 분포들이 고정된 형태(정규분포, 베르누이 분포 등)를 가지며, 그 형태가 파라미터 θ에 의해 결정된다는 것을 의미함. 즉, 막연한 함수가 아닌 θ로 표현되는 분포임.
2. Differentiablility:

- θ에 대해 미분 가능: 파라미터 θ를 조금 바꿨을 때 확률값이 부드럽게 변해야함. 그래야 역전파(경사하강법)를 통해 θ를 학습시킬 수 있음.
- z에 대해 미분 가능: 잠재 변수 z값이 조금 변할때도 확률값이 부드럽게 변해야 함. 이는 재매개변수화 트릭을 통해 z를 타고 gradient가 흘러가야 하기 때문에 필수적인 조건임.(만약 z가 discrete하다면 미분이 불가능해져서 VAE 작동 방식을 적용하기 어려움.)

> Q. 이때 확률값이 의미하는게 무엇인가?
> A. $p_\theta(x|z)$는 디코더가 출력하는 확률분포임. 잠재 변수 z가 주어졌을 때 데이터 x가 생성될 확률(밀도)임. 이때 부드럽게(Smoothly)변한다는 것은 수학적으로 미분 가능하다는 뜻임. 또, 이미지 도메인에서 고차원 내 두 점 사이 이미지가 두 이미지의 특성을 가지며 부드럽게 변함을 의미.

실제로 θ^*와 z^i에 대해 알 수 없음.

매우 중요하게도, marginal probabilities나 posterior probabilities에 대해 단순 가정을 하지 않음. 기존의 많은 모델은 계산을 쉽게 하기 위해 $p_\theta(x)$나 $p_\theta(z|x)$에 대해 특정한 형태(예: 정규분포)를 가정함. 하지만 VAE는 확률분포를 신경망으로 모델링하여 이러한 가정을 하지 않음으로 데이터가 매우 복잡한 분포를 따르거나, 잠재변수와 데이터 사이의 관계가 매우 복잡해도(비선형적이어도) 상관없다는 뜻임.

> Q. 주변확률이나 사후확률에 대해 단순 가정을 하지 않는다는 의미가 무엇인가?
> A. 확률 분포를 신경망으로 모델링하여 복잡한 분포를 근사할 수 있음.

아래는 기존 방법들로 해결하기 힘든 조건 2가지(Intractiablilty, Large dataset)를 제시함.

1. Intractability(난해성):

- p(x) 적분 계산 불가 -> 직접적인 MLE 불가능.
- p(z|x) 계산 불가 (베이지안 정리를 통해 도출된 분모 p(x)를 모름) -> EM 알고리즘 사용 불가능
- 평균 장 VB적분 계산 불가 -> 전통적인 VB(variational bayes) 사용 불가능

즉, 기존의 모든 표준적인 방법론이 막히는 상황임.
이런 intractable한 상황은 매우 흔하며, 적당히 복잡한 우도함수 $p_\theta(x|z)$, 예를 들어 비선형 은닉층(nonlinear hidden layer)을 가진 신경망 같은 경우에 나타남.(신경망은 비선형 함수이기 때문에, 이를 포함한 적분식은 손으로 풀 수 없기 때문임.)

2. large dataset: 데이터가 매우 커서 배치 최저화는 비용이 너무 많이 듦.
   이런 상황에서 SGD를 방식을 쓰고 싶음. 예를 들어 monte carlo EM과 같은 샘플링 기반 솔루션은 데이터포인트마다 비싼 샘플링 루프를 포함함.

위 시나리오에서 세가지 목표를 제시함.

- 생성모델(Decoder, $p_\theta(x|z)$)의 파라미터 θ를 최대 우도 추정치(MLE) 또는 MAP 추정치로 학습하여 데이터를 생성하는 것.
- 인식 모델(Encoder, $q_\phi(z|x)$)의 파라미터 ϕ를 학습하여 잠재 변수 z에 대한 근사 사후 분포를 추론하는 것.
- $p_\theta(x)$를 대략적으로라도 아는 것. 어떤 데이터 x가 '정상적인 데이터'인지 '이상한 데이터'인지를 판단하는 능력.

위 문제를 해결하기 위해, 실제 사후 분포 $p_\theta(z|x)$에 대한 근사(approximation)인 인식 모델(encoder)$q_\phi(z|x)$를 도입함.
전통적인 평균장 방식은 계산을 위해 모든 z가 독립적이라고 가정해야 하지만, VAE의 방식은 그런 가정을 하지 않음.(z끼리 상관관계가 있어도 학습 가능. iid가 필수 조건이 아니라는 뜻.)
전통적으로 최적의 파라미터를 찾기 위해 복잡한 적분 방정식을 풀어 닫힌 형태(colsed-form)을 유도해야 했지만 VAE는 딥러닝(SGD)으로 파라미터 ϕ를 점진적으로 학습함.
AEVB 알고리즘의 핵심은 추론(ϕ, 인코더)과 생성(θ, 디코더)을 별개의 과정으로 보지 않고, 하나의 목적 함수(ELBO)를 사용하여 동시에 최적화함.

> Q. 닫힌 형태(closed-form)란 무엇인가?
> A. 어떤 문제의 해답을 유한한 횟수의 표준적인 수학 연산과 수식만으로 정확하게 표현할 수 있는 해를 말함. 예를 들어 2차 방정식 $ax^2+bx+c=0$의 해는 근의 공식으로 한방에 구할 수 있음. 이것이 닫힌 해임.
> 닫힌 형태가 아닌 예로는 x = con(x)같은 방정식임. 반복적인 근사로 정답을 찾아야 함.

정보 이론에서 데이터를 전송하기 위해 압축하는 것을 Encoding, 다시 복원하는 것을 Decoding이라고 하는데, 이 개념을 차용함.
따라서 본 논문에서는 인식 모델 $q_\phi(z|x)$를 **확률적 인코더**(probabilistic encoder)라고 부를것임. 데이터 포인트 x가 주어졌을 때, 이 모델은 데이터 x를 생성했을 가능성이 있는 코드 값 **z에 대한 분포를 산출**하기 때문.(일반적인 오토인코더는 고정된 z벡터하나(deterministic)만 출력하지만 VAE는 z의 분포(평균과 분산)를 출력.)

비슷한 맥락에서 생성 모델 $p_\theta(x|z)$를 **확률적 디코더**(probabilistic decoder)라고 부름. 코드 z가 주어졌을 때, 이 모델은 가능한 대응 값 x에 대한 **분포를 산출**함.(z를 받아 x를 바로 뱉는게 아닌(결정론적X) x가 나올 확률 분포(베르누이 분포의 p값, 가우시안의 평균과 분산)를 출력하기 때문에 디코더가 정의한 확률 분포 함수 자체를 의미)
확률 밀도를 최대화(통계적 의미)하는 것이 픽셀간 오차를 최소화(실제 구현에서 MSE, CrossEntropy를 사용용)하는 것과 동치임.

2.2 The variational bound

목표는 전체 데이터셋이 모델에서 생성될 확률(우도)를 최대화하는 것임. 이때 데이터가 서로 독립(i.i.d.)이므로, 전체 로그 우도(주변 우도, marginal likelihood)는 각 데이터 $x^{(i)}$의 로그 우도의 합으로 표현됨.
$\log p_\theta(x^{(1)}, x^{(2)}, \ldots, x^{(N)}) = \sum_{i=1}^{N} \log p_\theta(x^i)$

이제부터 단일 데이터 포인트 x^i에 대해 식을 전개.

$$\log p_\theta(x^{(i)}) = D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z|x^{(i)})) + \mathcal{L}(\theta, \phi; x^{(i)})$$

- 좌변(log p_θ(x))은 우리가 최대화하고 싶은 값임(intractable함).
- 우변 첫째 항(D_{KL}): 근사 사후 분포 q와 실제 사후 분포 p 사이의 차이임.(실제 p를 모르니 계산 불가능)
- 우변 둘째 항($\mathcal{L}$): 변분 하한(ELBO)이라고 불리는 값임. 이 값은 q에 대한 기댓값으로 표현되며, 우리가 최대화할 수 있는 값임.(편향되지 않은 추정치임)

Q. 위 식의 유도 과정은 무엇인가?

A.

$\log p_\theta(x)$의 우변(right hand side, RHS)의 첫번째 항은 근사 사후 분포와 실제 사후 분포의 KL발산임. 두 확률 분포(q,p)가 얼마나 다른지 측정하는 지표임. 이 값은 0보다 크고(non-negative), 두 분포가 같을때 0이 됨.

이때 KL발산이 non-negative이므로, 우변 두번째 항$\mathcal{L}(\theta, \phi; x^{(i)})$은 데이터 포인트 i의 주변 우도에 대한 (변분) 하한이라고 불리며 다음과 같이 쓸 수 있음.

$KL \geq 0$ 이므로, $\log p(x) = KL + \mathcal{L} \geq \mathcal{L}$ 이 성립.

즉, $\mathcal{L}$은 $\log p(x)$보다 항상 작거나 같은 값(하한선)임. 따라서 $\mathcal{L}$을 최대화하는 것은 $\log p(x)$를 최대화하는 것과 동일한 효과를 가짐.

ELBO의 수학적 정의:

$$\log p_\theta(x^{(i)}) \geq \mathcal{L}(\theta, \phi; x^{(i)}) = \mathbb{E}_{q_\phi(z|x)}[-\log q_\phi(z|x) + \log p_\theta(x, z)]$$

$\mathbb{E}_{q_\phi(z|x)}[-\log q_\phi(z|x) + \log p_\theta(x, z)]$에서 $\log p_\theta(x, z)$를 $\log p_\theta(x^{(i)}|z)p_\theta(z)$로 쪼개지므로, 위 $\mathcal{L}$를 다시 정리하면 VAE의 손실함수 형태가 됨.

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

ELBO는 근사 사후 분포와 사전분포 사이 KL발산에 음수를 취한 것과 재구성 우도의 기댓값의 합으로 표현됨.

> Q. 사전분포 $p_\theta(z)$에 왜 θ가 붙은건가?
> A.논문에서는 VAE뿐만 아니라 다양한 모델에 적용가능한 이론을 전개하고 있음. 이론적으로 사전 분포$p(z)$의 파라미터도 학습 가능한 θ의 일부로 설정할 수 있음. 즉, "학습될 수도 있는 가능성"을 열어두기 위해 $p_\theta(z)$로 표기함.
> 논문의 실험 섹션에서 사전분포$p(z)$가 파라미터를 갖지 않는 $\mathcal{N}(0, I)$를 사용한다고 명시함.

ELBO의 첫번째 항은 인코더가 만든 z의 분포(q)가 사전 분포(p, 보통 정규 분포)와 비슷해지도록 강제함.
두번째 항은 z에서 다시 x를 복원했을 때 원본과 비슷할수록(우도가 높을수록)값이 커짐.

> Q. ELBO(Evidence Lower Bound, 증거 하한)란 무엇인가?
> A. 우리가 최대화하고자 하는 $\log p(x)$의 하한값을 의미. $\log p(x)$가 intractable하기 때문에, 대신 계산 가능한 하한값을 정의하여 최대화함. ELBO는 두 가지 항으로 구성되는데 직관적으로 잠재변수 z로 x를 얼마나 잘 복원했는가(reconstruction term), 인코더가 만든 분포 q가 사전 분포 p와 얼마나 가까운가(Regularization, KL divergence)를 측정함.

> Q. $\log p(x)$와 ELBO의 관계는 무엇인가?
> A. $\log p(x)$는 ELBO와 KL발산의 합임. 이때 KL발산은 non-negative이므로 ELBO는 log p(x)의 하한임. 즉, ELBO를 최대화 하는 것은 간접적으로 log p(x)를 최대화 하는 것과 같음.

> Q. $\log p_\theta(x)$에서 KL발산 항은 왜 등장하며, 이 KL발산 항을 삭제해도 되도 문제 없는가? 여기서 말하는 KL발산은 ELBO항 내 KL발산이 아닌 $\log p_\theta(x)$의 $D_{KL}(q_\phi(z|x)||p_\theta(z|x))$항임. 의미를 보면 추정한 가짜 사후 분포(q)와 진짜 사후 분포(p_{true})사이의 차이를 나타냄.
> A. 최적화 관점에서 목적함수를 들여다 보면 KL발산 항의 $p_\theta(z|x)$은 intractable하기 때문에 계산할 수 없음.
> 따라서 이 항을 직접적으로 다룰 수 없음. 목표를 "우도(log p(x))를 직접 최대화하는 것" 대신 "우도의 하한선(ELBO)을 최대한 끌어올리는 것"으로 바꿈.

> Q. $p_\theta(z|x)$은 신경망이기 때문에 직접 다룰 수 있지 않은가?
> A. θ가 붙어있다 = 신경망이 계산한다 라고 생각할 수 있지만 여기서 의미는 다름. $p_\theta(z|x)$는 신경망 θ가 '출력하는 값'이 아니라, 신경망 θ에 의해 수학적으로 '결정되는(implied) 값'인데, 계산이 불가능함.

> 1. θ는 디코더$p_\theta(x|z)$의 파라미터임.
> 2. $p_\theta(z|x)$은 디코더의 역방향임. 이걸 구하려면 베이즈 정리를 써야함.
>
> $$p_\theta(z|x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)}$$
>
> 3. 분자 $p_\theta(x|z)$는 디코더, $p_\theta(z)$는 가정한 $\mathcal{N}(0, I)$임으로 둘다 계산 가능함.
> 4. 분모를 계산하려면 가능한 모든 z에 대해 확률$p_\theta(x) = \int p_\theta(x|z)p_\theta(z)dz$을 다 더해야(적분해야)함. 이 복잡한 비선형 신경망(디코더)를 포함한 수식을 모든 실수 공간 전체의 z에 대해 적분하는 것은 수학적으로 불가능(intractable)함.
> 따라서 이 값을 직접 계산하는 대신, 별도의 신경망(인코더)을 만들어 $q_\phi(z|x)$로 근사하는 것임.

> Q. log p(x)를 최대화한다는 의미는 무엇인가?
> A. 모델이 실제 관찰된 데이터(x)를 가장 잘 설명하도록 만든다는 의미임. 모델이 추정한 확률분포에서 x가 나올 확률을 최대화한다는 뜻임.

최종 목표는 $\mathcal{L}$ 최대화하는 최적의 ϕ(인코더)와 θ(디코더)를 찾는 것임.

> Q. θ와 ϕ의 gradient가 수학적으로 어떻게 정의되는가?

θ는 ELBO안쪽 $\log p_\theta$만 미분하면 됨(Monte Carlo 샘플링으로 해결가능). 그러나 ϕ에 대한 gradient를 계산하는 것은 쉽지 않음. 왜냐하면 $\mathbb{E}_{q_\phi(z|x)}$에 대한 기댓값으로 표현되기 때문임.

> Q. θ와 ϕ에 대한 미분이란? 왜 ϕ에 대한 미분이 θ에 대한 미분보다 어려운가?

이 문제는 기존 강화학습 알고리즘으로 해결할 수 있지만 이 방식은 분산이 너무 큼. 이 말은 같은 파라미터에서 계산할 때마다 gradient 방향이 제각각이라 학습이 불안정함. 이의 대안으로 재매개변수화 트릭을 제안.

2.3 The SGVB estimator and AEVB algorithm

앞서 제기된 '미분 어려움'과 '높은 분산'문제를 해결하기 위해 재매개변수화 트릭과 이를 적용한 알고리즘을 제시.
현재는 $q_\phi(z|x)$형태의 근사 사후 분포를 가정하지만, $q_\phi(z)$인 x에 조건부이지 않은 경우(인코더가 없는 경우)에도 적용될 수 있음.

2.4절에 기술된 조건 하에 $q_\phi(z|x)$에서 z를 샘플링하는 과정을 $\tilde{z} \sim q_\phi(z|x)$를 (보조) 노이즈 변수 ϵ의 미분 가능한 변환 $g_\phi(\epsilon, x)$를 사용하여 재매개변수화(reparameterize)할 수 있음.

기존에 $\tilde{z}$를 q_ϕ 분포에서 직접 뽑아서 미분이 끊김.
그래서 $\tilde{z}$를 함수 $g_\phi(\epsilon, x)$로 표현함. ϵ은 외부의 간단한 분포에서 샘플링하여 무작위성은 ϵ로 격리되고, $\tilde{z}$는 ϕ에 대해 미분 가능한 함수가 됨.
구체적인 방법은 2.4절에서 다룸.
$\mathbb{E}_{q_\phi(z|x^{(i)})}[f(z)] = \mathbb{E}_{p(\epsilon)}[f(g_\phi(\epsilon, x^{(i)}))] \simeq \frac{1}{L}\sum_{l=1}^{L} f(g_\phi(\epsilon^{(l)}, x^{(i)}))$ where $\epsilon^{(l)} \sim p(\epsilon)$
좌변은 원래 구하려던 기댓값(미분 불가능한 원본 문제). 중간은 z를 $g(\epsilon, x^{(i)})$로 치환(트릭)하여 기댓값의 대상이 $q(z)$에서 $p(\epsilon)$로 바뀜.(미분 가능해짐, 적분 변수가 ϵ으로 바뀜) 우변은 몬테 카를로 샘플링. 이때 ϵ은 ϕ와 무관하므로 미분 연산자가 시그마 안으로 들어갈 수 있음.
이 기술을 ELBO에 적용하여 SGVB 추정치 $\tilde{\mathcal{L}}(\theta, \phi; x^{(i)})$를 얻음.
$\tilde{\mathcal{L}}(\theta, \phi; x^{(i)}) = \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(x^{(i)}, z^{(i,l)}) - \log q_\phi(z^{(i,l)}|x^{(i)})$
위 식은 ELBO의 모든 항을 몬테 카를로 샘플링으로 계산하는 방식이며, 이때 z는 아래와 같이 정의되어 ϕ에 대해 미분 가능한 함수 g로 표현됨.
where $z^{(i,l)} = g_\phi(\epsilon^{(i,l)}, x^{(i)})$ and $\epsilon^{(l)} \sim p(\epsilon)$

VAE의 성능향상:
앞서 추정한 A는 모든 항을 샘플링으로 계산함. 이때 q(인코더의 출력)와 p(prior)가 모두 가우시안이라면, KL발산은 굳이 불안정한 샘플링을 하지 않아도 수학 공식으로 정확한 값을 바로 구할 수 있음. 샘플링(무작위성)을 줄이고, 정확한 계산(수식)으로 대체하면 추정치의 분산이 감소하여 학습이 더 안정적이게 됨.

KL 발산 항은 근사 사후 분포가 사전 분포에 가까워지도록 장려하면서 ϕ를 규제하는 것으로 해석될 수 있음.
$\tilde{\mathcal{L}}(\theta, \phi; x^{(i)}) = -D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z)) + \frac{1}{L}\sum_{l=1}^{L}(\log p_\theta(x^{(i)}|z^{(i,l)}))$
where $z^{(i,l)} = g_\phi(\epsilon^{(i,l)}, x^{(i)})$ and $\epsilon^{(l)} \sim p(\epsilon)$
이 두번째 SGVB 추정치$\tilde{\mathcal{L}}$는 가 $\tilde{\mathcal{L}}$보다 더 적은 분산을 가진다.

> Q. 왜 두번째 SGVB추정치 $\tilde{\mathcal{L}}$가 $\tilde{\mathcal{L}}$보다 더 적은 분산을 가지는가?
> A.무작위 샘플링을 줄이고, 정확한 수식으로 대체했기 때문임.
> $\tilde{\mathcal{L}}$은 p나 q가 가우시안이 아니어도(수학 공식을 몰라도)쓸 수 있는 **범용적**인 방법임.
> $\tilde{\mathcal{L}}$은 p와 q가 공식이 명확한(가우시안 등) 분포로 가정되었을 경우에 KL발산은 적분 없이 깔끔하게(무작위성이 전혀 없는 상수처럼) 계산됨(**효율적**).

> Q. q와 p가 가우시안일 때 KL발산을 수식으로 바로 구할 수 있는 이유는 무엇인가? 또, KL발산을 수식으로 구하는 것이 샘플링보다 왜 더 안정적인가?
> A.가우시안 분포의 로그를 취하면 2차 함수 형태가 되어 적분이 매우 쉬워짐.
> KL 발산은 기본적으로 적분임. $D_{KL}(q||p) = \int q(z)(\log q(z) - \log p(z))dz$
> 이 적분을 풀려면 모든 z에 대해 넓이를 구해야 하는데 가우시안 분포는 지수 함수의 형태임.
>
> $$q(z) = \frac{1}{2\pi\sigma}e^{-\frac{(z-\mu)^2}{2\sigma^2}}$$

때문에 로그를 취하면 log와 e가 만나 사라지고, 지수에 있던 2차식이 분자로 감.

$$\log q(z) \approx -\frac{(z-\mu)^2}{2\sigma^2} + \text{상수}(= \frac{1}{2\pi\sigma})$$

이로써 문제가 가우시안 분포 하에서 2차 식의 평균(기댓값)을 구하라는 문제로 바뀜. 가우시안 분포의 평균(μ)과 분산(σ^2)의 정의 자체가 1차, 2차 식에 대한 기댓값이므로, 복잡한 적분 없이 μ와 σ만의 사칙연산으로 답이 나옴.

$$D_{KL} = -\frac{1}{2}\sum_{j=1}^{J}(1 + \log((\sigma_j)^2) - (\mu_j)^2 - (\sigma_j)^2)$$

적분 기호가 사라지고, 단순한 사칙연산만 남음. 이것이 닫힌 형태임.
다시 질문으로 돌아오면 수식으로 구하는 것이 샘플링보다 왜 더 안정적인가? 위 유도과정에 따라서 무작위성(샘플링)이 제거되 분산이 0이 됨. 논문은 가능한 경우(p와 q를 가우시안으로 가정하는 경우)에는 반드시 수식을 사용하는 추정치 B를 권장하며, 이것이 실제로 학습 속도와 안정성을 크게 높여줌.

$\tilde{\mathcal{L}}$의 첫번째 항($-D_{KL}$): 해석적 계산(공식 사용, 샘플링 X)
두번째 항($\sum \log p$): 재구성 오차, 재매개변수화 트릭 사용(샘플링 O)

$\mathcal{L}(\theta, \phi; X) \simeq \tilde{\mathcal{L}}(\theta, \phi; X) = \frac{N}{} \sum_{i=1} \tilde{\mathcal{L}}(\theta, \phi; x^{(i)})$
딥러닝 학습을 위해 전체 데이터 N개 중 M(minibatch)개를 뽑아 평균을 구하고, 다시 N을 곱해 전체 크기로 스케일링하는 방식임.
$\frac{N}{}$ 항이 이 미니배치 합을 전체 데이터셋 크기로 스케일링(Scaling)해주기 위함.

이론적으로 기댓값을 정확히 계산하려면 z를 여러번(L번) 뽑아서 평균을 내야 하지만 미니배치 학습(SGD)에서는 데이터가 100개(M=100) 모여 있으므로, 각 데이터마다 z를 1번만 뽑아도(L=1), 전체적으로 100번의 샘플링 효과가 발생하여 노이즈가 상쇄됨.

도함수 $\nabla_{\theta,\phi}\tilde{\mathcal{L}}(\theta; X)$를 취할 수 있고, 결과적인 gradient는 SGD나 Adagrad와 같은 optimizer로 함께 사용될 수 있음. -> 재매개변수화 트릭 덕분에 미분이 가능해졌으므로, 딥러닝에서 흔히 쓰는 옵티마이저를 그대로 사용할 수 있다는 뜻임.

Algorithm 1

Algorithm 1 Minibatch version of the Auto-Encoding VB (AEVB) algorithm. Either of the two SGVB estimators in section 2.3 can be used. We use settings $M = 100$ and $L = 1$ in experiments.

$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Initialize parameters
repeat
  $\mathbf{X}^M \leftarrow$ Random minibatch of M datapoints (drawn from full dataset)
  $\boldsymbol{\epsilon} \leftarrow$ Random samples from noise distribution $p(\boldsymbol{\epsilon})$
  $\mathbf{g} \leftarrow \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \tilde{\mathcal{L}}^M(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{X}^M, \boldsymbol{\epsilon})$ (Gradients of minibatch estimator (8))
  $\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Update parameters using gradients $\mathbf{g}$ (e.g. SGD or Adagrad [DHS10])
until convergence of parameters $(\boldsymbol{\theta}, \boldsymbol{\phi})$
return $\boldsymbol{\theta}, \boldsymbol{\phi}$

1. $\theta, \phi \leftarrow$ Initialize parameters: 인코더(ϕ)와 디코더(θ) 신경망의 가중치를 초기화(보통 랜덤 값으로).
2. Repeat: 학습 루프 시작 (수렴할 때까지 반복).
3. $X \leftarrow$ Random minibatch of M datapoints (drawn from full dataset): 전체 데이터에서 미니배치 M개를 뽑음
4. $\epsilon \leftarrow$ Random samples from noise distribution $p(\epsilon)$: 재매개변수화에 쓸 노이즈 ϵ를 샘플링.(보통 표준정규분포 $\mathcal{N}(0,1)$에서 개 뽑음)
5. $g \leftarrow \nabla_{\theta,\phi} \tilde{\mathcal{L}}(\theta, \phi; X, \epsilon)$ (Gradients of minibatch estimator (8))
   - 인코더 Forward: $x \rightarrow \mu, \sigma$
   - 재매개변수화: $z = \mu + \sigma \odot \epsilon$
   - 디코더 Forward: $z \rightarrow \text{econstrction}$
   - Loss 계산: $\text{econstrction Loss} + \text{KL Divergence}$
   - Backprop: Loss를 θ, ϕ로 미분하여 그래디언트 g 계산.
6. $\theta, \phi \leftarrow$ Update parameters using gradients g: 계산된 g로 가중치 업데이트
7. Until convergence of parameters (θ, ϕ): 학습 종료 조건
8. Return θ, ϕ: 학습된 모델 반환.

ELBO의 KL발산 항은 근사사후분포$q_\phi(z)$와 사전분포$p(z)$ 사이의 **규제**(인코더가 데이터x를 잠재공간으로 보낼 때, 그 분포가 정규분포 모양을 갖추도록 강제)역할이며, 다른 항$\log \mathrm{p}_\theta(x^{(i)}|z)$은 z에서 x로 복원했을 때 확률 밀도를 최대화함.(음의 재구성 오차: z를 통해 예측된 $\tilde{x}$와 실제x의 차이, MSE)

인코더와 재매개변수화 트릭이 $x \rightarrow z$ 매핑(인코딩)과정을 담당함.
$z^{(i,l)} = g_\phi(\epsilon^{(l)}, x^{(i)})$ 여기서 $z^{(i,l)} \sim q_\phi(z|x^{(i)})$

그 후 샘플 $z^{(i,l)}$는 $\log p_\theta(x^{(i)}|z^{(i,l)})$에 입력되는데 이는 $z^{(i,l)}$가 주어졌을 때 데이터 $x^{(i)}$의 확률 밀도(또는 질량)와 같음.(생성모델이(디코더) $z \rightarrow x$ 매핑 과정을 담당.)
결론적으로 VAE의 학습 과정은 입력을 압축(q)했다가 복원(p)하면서 오차를 줄이는 것(음의 재구

성 오차)이라는 오토인코더와 수학적으로 동일함. 다만 확률적인 규제(KL term)가 추가되어, 잠재 공간이 정규분포와 비슷한 형태를 갖추도록 강제함.

2.4 The reparameterization trick

연속 확률 잠재 변수 z를 조건부 분포 $z \sim q_\phi(x|z)$에서 직접 샘플링하는 대신, 미분이 가능하도록 우회하는 대안을 사용.
확률 변수 z를 확률(ϵ)과 함수($g(\phi)$) 두 부분으로 쪼갬.
ϵ: 파라미터 ϕ와 무관한 순수 노이즈.($\epsilon \sim p(\epsilon)$)
g_ϕ: ϕ를 포함하는 미분 가능한 껍데기 함수.

이 트릭을 쓰는 유일한 목적은 ϕ에 대한 미분 가능성(backpropagation) 확보임을 강조.
결정론적 매핑 $z = g_\phi(\epsilon, x)$가 주어졌을 때, 우리는 $q_\phi(z|x) \prod_i dz_i = p(\epsilon) \prod_i d\epsilon_i$임.

> Q. 이게 도대체 무슨 말임???????????
> A.

z공간에서의 확률 질량q(z)dz 과 매핑된 ϵ공간에서의 확률 질량 $p(\epsilon)d\epsilon$은 보존되어야 한다는 원리임.
따라서, $\int q_\phi(z|x) f(z) dz = \int p(\epsilon) f(z) d\epsilon = \int p(\epsilon) f(g_\phi(\epsilon, x)) d\epsilon$임.
좌변: 원래 구하려던 적분. 적분 변수가 z이고, 확률 밀도q_ϕ가 ϕ에 의존함.(미분 난해)
우변: 치환된 적분. 적분 변수가 ϵ이고, 확률 밀도$p(\epsilon)$는 ϕ와 무관함. ϕ는 피적분 함수 $f(g_\phi)$안으로 들어감.
이제 적분 기호($\int$)나 확률 분포(p)가 ϕ에 영향을 받지 않으므로, 미분 연산자 ∇_ϕ가 적분 안으로 자유롭게 들어갈 수 있음.

> Q. 아니 이게 대체
> A.

결과적으로 미분 가능한 추정치가 구성됨. 적분을 몬테카를로 합으로 바꾼 형태임.
예를 들어 일변량 가우시안의 경우($z \sim p(z|x) = \mathcal{N}(\mu, \sigma^2)$)에서 재매개변수화는 $z = \mu + \sigma\epsilon$이며, 여기서 ϵ은 보조 노이즈 변수$\epsilon \sim \mathcal{N}(0, 1)$임.
가우시안 말고 또 어떤 분포에 이 트릭을 쓸 수 있는지 확장성을 설명함.
1. 계산 가능한 역 CDF(누적 분포 함수). 이 경우$\epsilon \sim (0, I)$(0과 1 사이의 균등 분포)로 두고, $g_\phi(\epsilon, x)$를 $q_\phi(z|x)$의 역 CDF로 설정함. CDF의 역함수만 알면, 0~1사이의 난수(ϵ)를 꽂아 넣어 원하는 분포의 샘플 z를 만들 수 있음.Examples: Exponential, Cauchy, Logistic, Rayleigh, Pareto, Weibull, Reciprocal, Gompertz, Gumbel and Erlang distributions.
2. 가우시안과 유사하게 모든 위치-스케일 분포족에 대해 표준 분포(위치 0, 스케일 1)를 보조 변수 ϵ으로 선택하고, $g(.) =$ 위치+스케일·ϵ으로 설정할 수 있음.(모양은 똑같은데, 평균과 크기만 바뀌

는 분포들이 모두 사용가능하다는 것임.) Examples: Laplace, Elliptical, Student's t, Logistic, Uniform, Triangular and Gaussian distributions.
3. 합성:확률 변수를 보조 변수들의 서로 다른 변환으로 표현하는 것도 가능함. 예: 로그-정규(정규 분포 변수의 지수화), 감마(지수 분포 변수들의 합), 디리클레(감마 변수들의 가중 합), 베타, 카이 제곱, F 분포.
결론적으로, 연속 확률 변수라면 거의 대부분 이 재매개변수화 트릭을 적용하여 VAE로 학습시킬 수 있다는 강력한 범용성을 주장.

3. Example: Variational Auto-Encoder

신경망을 확률적 인코더로 사용하고, 파라미터 ϕ, θ가 AEVB알고리즘으로 공동 최적화되는 예시를 듦.(이론적 q, p를 구체적인 신경망으로 구현)

잠재 변수에 대한 사전 분포(prior)를 중심화(평균=0)된 등방성(공분산 행렬이 항등행렬임. 즉, 잠재 변수의 각 차원은 서로 독립이며 분산이 1) 다변량 가우시안 $p_\theta(z) = \mathcal{N}(z; 0, I)$로 둠.
이 경우에 사전 분포$p(z)$에는 (학습할) 파라미터가 없음.(일반화하기 위해 $p(z)$를 $p_\theta(z)$로 표기한 것.)
디코더 $p_\theta(x|z)$는 z를 입력받아 x의 분포 파라미터를 뱉는 MLP임. 이때 데이터(x) 타입에 따라 MSE(가우시안)을 쓸지, BCE(베르누이)를 쓸지 결정됨.
이때 디코더가 복잡한 비선형 신경망이므로, 역방향인 **실제 사후 분포** $p_\theta(z|x)$는 계산 불가능함.(베이즈 정리를 따라가보면 됨.)
인코더 $q_\phi(z|x)$의 형태에는 많은 가정을 할 수 있지만, 실제(다루기 힘든) 사후 분포가 대략적으로 **대각 공분산**을 갖는 **가우시안** 형태를 띤다고 가정할 것이고, 이 경우에 변분 근사 사후분포(q)를 대각 공분산 구조를 가진 다변량 가우시안으로 설정할 수 있음.

$$\log q_\phi(z|x^{(i)}) = \log \mathcal{N}(z; \mu^{(i)}, \sigma^{2(i)} I)$$

여기서 근사 사후 분포의 평균μ과 표준편차σ는 인코딩 MLP의 출력임. 즉, 데이터 포인트 $x^{(i)}$와 변분 파라미터 ϕ의 비선형 함수임.

섹션 2.4에서 설명(재매개변수화 트릭)한대로 $z^{(i,l)} = \mu^{(i)} + \sigma^{(i)} \odot \epsilon^{(l)}$ (여기서 $\epsilon^{(l)} \sim \mathcal{N}(0, I)$)를 사용하여 사후 분포에서 $z^{(i,l)}$을 샘플링함.
이때 사전 분포와 근사 사후 분포(인코더: $q_\phi(z|x)$)는 모두 가우시안이기 대문에 KL 발산이 추정(샘플링) 없이 계산되고 미분될 수 있는 추정치 B를 사용하여 닫힌 형태를 구할 수 있음.

이 모델과 데이터 포인트 $x^{(i)}$에 대한 결과적인 추정치(Loss function)은 다음과 같음.

$$\mathcal{L}(\theta, \phi; x^{(i)}) \simeq \frac{1}{2} \sum_{j=1}^{J} (1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2) + \frac{1}{L} \sum_{l=1}^{L} \log p_\theta(x^{(i)}|z^{(i,l)})$$

첫번째 항: KL 발산 항(regularization)

샘플링(z)이 전혀 안 들어감. 오직 인코더 출력인 μ, σ로만 계산됨.
이 식은 부록 B에서 유도된 가우시안 간의 KL 발산 공식에 마이너스를 붙인 것.
J: 잠재 변수 z의 차원 수.
두번째 항: 재구성 오차 항(reconstruction)
여기에는 샘플링된 $z^{(i,l)}$이 들어감.

$$z^{(i,l)} = \mu^{(i)} + \sigma^{(i)} \odot \epsilon^{(l)}, \epsilon^{(l)} \sim \mathcal{N}(0, I)$$

디코더 $\log p_\theta(x^{(i)}|z^{(i,l)})$는 모델링하는 데이터 타입(실수, 이진)에 따라 가우시안 또는 베르누이 MLP임.
VAE가 꼭 가우시안이나 베르누이만 써야 하는건 아니며, 필요하면 다른 복잡한 분포를 써도 되는 유연함을 가짐.

> Q. 인코더와 디코더의 출력은 모두 각 차원의 평균과 분산벡터인가?
> A. 인코더의 출력은 평균 벡터μ와 분산 벡터 σ임.
> 디코터의 출력도 확률 분포의 **파라미터 (μ_x, σ_x) 두 벡터가 출력됨. $x \sim \mathcal{N}(\mu_x, \sigma_x^2 I)$로써 샘플링된 x가 엄밀하게 복원된 x지만, 실제로는 노이즈(σ_x)를 무시하고 평균값(μ_x) 자체를 바로 이미지로 사용하는 경우가 대부분임. μ_x가 가장 확률이 높은 대푯값이기 때문.
> 학습 관점에서는 디코더가 출력한 μ_x, σ_x와 원본 이미지 x를 수식(가우시안 로그 우도 식)에 대입하여 확률을 계산하는데 이 과정은 음의 재구성 오차(MSE)와 동일함.

4. Related Work

VAE와 가장 유사하다고 알려진 Wake-Sleep 알고리즘(이후 WS로 표기)은 데이터가 들어오는 대로 학습(실시간, 온라인)이 가능하고, 연속적인 z를 다룰 수 있다는 점이 VAE와 같음.
WS도 VAE의 인코더처럼 $x \to z$를 추론하는 별도의 모델은 둠으로써 구조적으로 VAE와 매우 흡사함.
WS 알고리즘의 단점으로는 두 개의 목적 함수를 동시에 최적화해야 한다는 것인데, 이 두개가 합쳐졌을 때 주변 우도(의 하한)의 최적화와 일치하지 않는다는 것임. VAE는 ELBO만 최대화하여 수학적으로 $\log p(x)$도 높아진다는 보장이 있지만 WS는 'Wake' 단계와 'Sleep'단계에서 사용하는 목적 함수가 서로 다름.(KL 발산의 방향이 반대임: KL(q||p) vs KL(p||q)). 때문에 WS 알고리즘이 수렴하더라도, 그것이 진짜 정답 $\log p(x)$를 최대화한다는 수학적 보장이 없음.
WS의 장점으로는 이산 잠재 변수를 가진 모델에도 적용됨. VAE는 미분(gradient)을 해야 하기에 z가 연속적이어야 하지만, WS은 샘플링 기반이라 z가 0이나 1같은 이산 값이어도 작동함.(물론 VAE도 나중에 Gumbel-Softmax 등으로 이를 극복함.)
WS은 데이터 포인트당 계산 복잡도가 AEVB와 동일함.

확률적 변분 추론(Stochastic Variational Inference, SVI)이 최근 관심이 증가함.
VAE는 '재매개변수화'로 분산을 잡았지만, 다른 연구자들은 제어 변량(Control Variate)이라는 통계적 기법으로 분산을 잡으려고 함. [RGB13]에서 그래디언트 추정치의 분산을 줄이기 위한 몇 가지 일반적인 방법, 즉 제어 변량 기법이 소개됨.

[SK13]에서 지수족 근사 분포의 자연 파라미터를 학습하기 위한 효율적인 확률적 변분 추론 알고리즘에서 VAE와 유사한 재매개변수화가 사용됨. VAE는 이를 일반적인 딥러닝 신경망에 적용했다는 점이 차이점임.

Wake-Sleep: HDFN95
SVI: HBWP13
제어 변량 기법: BJP12, RGB13
VAE와 유사한 재매개변수화: SK13

AEVB알고리즘은 (변분 목적 함수로 훈련된) 방향성 확률 모델과 오토인코더 사이의 연결고리를 드러냄.
선형 오토인코더(PCA)와 특정 종류의 생성 선형-가우시안 모델 사이의 연결은 예전부터 존재했음. [Row98]에서는 PCA가 사전 분포 $p(z) = \mathcal{N}(0, I)$와 조건부 분포 $p(x|z) = \mathcal{N}(x; Wz, \epsilon I)$를 갖는 선형-가우시안 모델의 특수한 경우, 구체적으로는 ϵ이 무한히 작은(infinitesimally small) 경우에 확률적인 성격이 사라지고 결정론적인 일반 PCA가 됨으로써 최대 우도(ML) 해에 해당한다는 것이 보여졌습니다. VAE는 이 PCA의 아이디어를 비선형(신경망)으로 확장하고, 노이즈(ϵ)를 0으로 없애지 않고 확률적 성질을 살려둔 일반화된 버전이라고 볼 수 있음.

> Q. 이 상황에서 최대 우도 해란?
> A. 일반적으로 PCA(주성분분석)는 데이터의 분산을 최대화하는 축을 찾는 '기하학적' 방법으로 알려져 있음. 이를 확률적 모델(Probabilistic)로 해석할 수 있음.
> 선형-가우시안 모델: $x = Wz + \mu + \epsilon (z \sim \mathcal{N}(0, 1), \epsilon \sim \mathcal{N}(0, \sigma^2 I)$
> 최대 우도 해(ML Solution): 이 확률 모델에서 관측 데이터 X가 등장할 확률($likelihood, p(X|W, \sigma^2)$)을 최대화하는 파라미터 W를 찾는 것을 의미.
> 이 확률 모델에서 노이즈 ϵ(또는 σ^2)을 0으로 수렴시키면 모델이 확률적인 성격을 잃고 결정론적이 됨. 이때 우도를 최대화하는 W를 구해보면, 전통적인 PCA가 구하는 고유벡터와 수학적으로 정확히 일치함.
> 여기서 **최대 우도 해**란 데이터의 확률 밀도를 가장 높게 만드는 최적 파라미터 W를 뜻하며, ϵ이 0에 수렴하는 상황에서 이것이 곧 PCA의 해와 같아진다는 뜻임.
> "이 부분은 더 봐야할듯..."

오토인코더에 관한 최근 연구[VLL+10]에서 정규화되지 않은 오토인코더의 훈련 기준은 입력 x와 잠재 표현 z사이의 상호 정보량(mutual information)의 하한을 최대화하는 것과 대응된다는 것이 보여짐. 기존 오토인코더도 나름의 확률적 해석(상호 정보량 최대화)이 있었다는 것.

> Q. 입력 x와 잠재 표현 z사이의 상호 정보량의 하한을 최대화하는 것이 무슨 의미인가?
> A. 상호정보량 $I(X; Z)$: 변수 Z를 알았을 때 변수 X에 대한 불확실성이 얼마나 줄어드는지를 나타내는 척도임. 즉, Z가 X에 대한 정보를 얼마나 많이 담고 있는가를 의미함. 좋은 오토인코더는 $I(X; Z)$를 최대화하는 모델임. 하지만 이 상호 정보량을 직접 계산하는 것은 매우 어렵고, 수학적으로 유도해보면 음의 재구성 오차가 상호 정보량의 하한이 됨을 알 수 있음. 즉, 재구성 오

> 차(MSE)를 최소화하는 작업이 수학적으로는 입력 X와 잠재 변수 Z 사이의 정보 공유량(상호 정보량)의 하한선을 최대한 끌어올려서 정보를 많이 담으려는 것과 동치라는 것임.

오토인코더가 단순히 입력을 베끼는게 아닌, 정보 이론적으로 입력 x와 잠재 변수 z사이의 정보 공유량(상호 정보량)을 최대화하는 과정임을 설명함. 수학적으로 풀면 상호 정보량을 최대화하려는 것이 재구성 오차를 줄이는 것과 연결됨. 즉, 복원 오차를 줄이는 기존 오토인코더의 학습 방식이, 사실 x의 정보를 z에 최대한 많이 담으려는 정보 이론적 행동임을 설명함.

> Q.오토인코더를 정보이론적 관점에서 해석하면 무슨 의미인가?
> A. 오토인코더의 학습은 데이터 압축임. 정보이론적 목표는 제한된 용량(Bottleneck, 저차원 Z)을 가진 공간을 통과하면서 정보 손실을 최소화하는 것. 이때 재구성 오차를 줄인다는 것은, 압축된 코드 Z만 있어도 원본 X를 거의 완벽하게 설명할 수 있도록 가장 효율적인 정보 전달 방법을 배우는 과정임.

그러나 오토인코더가 단순히 입력을 그대로 출력으로 복사만 한다면 데이터를 압축하거나 특징을 배우는 의미가 없음.
오토인코더가 유용한 표현을 학습하게 만들기 위해 디노이징(Denoising), 수축(Contractive), 희소(Sparse, 가중치 0) 오토인코더 변형과 같은 정규화 기법(인위적인 규제)들이 제안됨.
위 오토인코더들은 노이즈를 얼마나 줄지, 얼마나 희소하게 할지 같은 하이퍼 파라미터를 사용자가 조절해야 했지만, VAE의 SGVB 목적함수는 변분 하한에 의해 결정되는 정규화 항(KL 발산)을 포함하고 있어서, 유용한 표현을 학습하기 위해 필요한(성가신) 정규화 하이퍼파라미터가 없음.
PSD(Predictive Sparse Decomposition)[KRL08]와 같은 인코더-디코더 구조들도 관련이 있음.
최근 소개된 GSN(Generative Stochastic Networks)[BTL13]은 디노이징 오토인코더가 데이터 분포에서 샘플링하는 마르코프 제인 전이 연산자를 학습하는 방식임.
[SL10]에서 심층 볼츠만 머신(Deep Boltzmann Machines, DBM)의 효율적인 학습을 위해 인식 모델(x를 보고 z를 추론하는 별도의 모델)이 사용됨. 하지만 DBM은 비방향성 모델인 반면, VAE는 $z \rightarrow x$의 인과 관계가 명확한 방향성(Directed)모델이라는 점에서 차이가 있음.
이런 방법들은 정규화되지 않은 모델(ex. 볼츠만 머신)을 목표로 하거나, 희소 코딩 모델에 국한되는 반면, VAE는 일반적인 범주의 방향성 확률 모델을 학습한다는 점에서 대조됨.

당시 최근 제안된 DARN[GMW13] 또한 오토인코딩 구조를 사용하여 방향성 확률 모델을 학습하지만, 이 방법은 이진 잠재 변수에 적용됨.
그보다 더 최근에 [RMW14]는 VAE의 재매개변수화 트릭을 사용하여 오토인코더, 방향성 확률모델, 확률적 변분 추론사이의 연결고리를 만듦.
다른 연구들과 독립적이지만 같은 결론에 도달함.

5. Experiments

mnist와 frey face 데이터셋으로 여러 생성모델을 훈련. 변분 하한과 추정된 주변 우도(p(x), 모델이 실제 데이터를 얼마나 잘 설명하는지 추정값.) 측면에서 각 알고리즘을 비교.

> Q. 훈련 관점에서 변분 하한과 주변 우도가 가지는 의미가 무엇인가?
> A. 주변 우도 $\log p_\theta(x)$: 모델이 데이터를 얼마나 그럴듯하게 생각하는가에 대한 진짜 점수. 이상적인 목표는 이 값을 최대화하는 건데 intractable하여 직접 최적화에 사용할 수 없음.
> 변분하한$\mathcal{L}$: $\log p_\theta(x)$보다 항상 작거나 같은 값. $\log p(x) \geq \mathcal{L}$ 훈련 관점에서 직접 계산하고 미분할 수 있는 실질적인 손실 함수임. 훈련 과정은 이 하한선을 최대한 위로 밀어 올리는(maximize) 과정임. 하한선이 올라가면 주변우도도 같이 올라갈 가능성이 높기 때문. 또한, ELBO는 재구성오차와 정규화(KL)의 균형을 맞추는 역할도 함.

"The generative model (encoder) and variational approximation (decoder) from section 3 were used, where the described encoder and decoder have an equal number of hidden units."
`원문에 생성모델(인코더), 변분 근사(디코더)라고 정의되어 있는데 오타인듯.`
변분 근사(인코더, q), 생성모델(디코더, p) 두 신경망의 크기(hidden units, 은닉층을 의미)를 똑같이 맞춤.
frey face 데이터는 연속적(픽셀 값이 0~1 사이 실수값)이므로, 디코더 마지막 층에 sigmoid를 씌워서 출력값(μ_x)이 0과 1 사이가 되도록 강제하고, 이를 평균으로 하는 가우시안 분포를 사용함(MSE Loss 사용).

파라미터는 하한 추정치 $\nabla_{\theta,\phi}\mathcal{L}$를 미분하여 계산된 그래디언트와 사전 분포 $p(\theta) = \mathcal{N}(0, I)$에 해당하는 작은 가중치 감쇠(weight decay)항을 더하여 SGD로 업데이트함(최적화). weight decay란 파라미터 θ가 너무 커지지 않도록 억제하는 정규화 기법(L2 Regularization)을 썼는데, 이는 베이지안 관점에서 파라미터 θ에 대한 사전 분포를 가우시안으로 둔 것과 수학적으로 같음.
이 목적 함수 최적화는 근사 MAP 추정과 등가이며, 여기서 우도 그래디언트$\nabla_\theta p_\theta(x)$는 하한의 그래디언트$\nabla_{\theta,\phi}\mathcal{L}$로 근사됨. 파라미터에 대한 사전 분포$p_\theta(x)$까지 고려했으므로, 단순 최대 우도(Maximum Likelihood)가 아닌 MAP 추정임.

AEVB와 Wake-Sleep과 비교할 때 모델 구조는 똑같이 맞추고, 학습 알고리즘만 다르게 적용함.
ϕ, θ 모두 $\mathcal{N}(0, 0.01)$에서 무작위 샘플링하여 초기화. MAP기준을 사용하여 공동으로 확률적으로 최적화됨.

> Q. MAP기준을 사용하여 공동으로 확률적으로 최적화됨이 무슨 의미인가?
> A. 두 파라미터가 따로 학습되지 않고(ex. GAN) 하나의 목적 함수(ELBO)를 사용하여 동시에 업데이트한다는 뜻임. 이때 '확률적'의 의미는 전체 데이터를 한 번에 쓰지 않고 미니배치를 사용하여 그래디언트를 추정하고 업데이트(SGD)한다는 뜻임.
> MAP기준:
> 일반적인 최대 우도(Maximum Likelihood)는 데이터$p(X|\theta)$만 최대화함.
> MAP는 파라미터 자체의 확률인 사전 분포 $p(\theta)$까지 고려하여 $p(\theta|X) \propto p(X|\theta)p(\theta)$를 최대화함.
> 논문에서 파라미터에 대한 사전 분포를 표준 가우시안으로 가정했기에 목적함수에 가중치 감쇠항이 추가됨.

> 즉, ELBO + Weight Decay(prior)를 최대화하는 것이 곧 MAP 추정을 근사적으로 수행하는 것과 같다는 의미임.

학습률은 Adagrad[DHS10]로 조정됨.Adagrad의 전역 스텝 사이즈 파라미터는 처음 몇 번의 반복에서 훈련 세트의 성능을 기반으로 {0.01, 0.02, 0.1} 중에 선택됨.
미니배치 M = 100, 데이터 포인트당 L=1개의 샘플을 사용함.

Likelihood Lower Bound (ELBO 비교)
mnist는 500개의 은닉 유닛, frey face(작은 데이터셋이므로 과적합 방지를 위해) 200개의 은닉 유닛을 가진 생성모델(디코더)와 인코더(인식 모델)를 훈련시킴.
선택된 은닉 유닛 수는 오토인코더에 대한 이전 문헌을 기초로 함. 알고리즘간의 상대적 성능은 이런 선택에 크게 민감하지 않았음.
Figure 2는 변분 하한(ELBO) 최적화 성능 비교: x축이 훈련 샘플 평가 횟수. 로그스케일임. 즉, 학습 시간 또는 반복 횟수를 의미. y축은 추정된 평균 변분 하한. 값이 높을수록 좋음. N은 잠재 변수 z의 차원 수임. AEVB가 Wake-Sleep보다 위에 있음 = VAE가 훨씬 더 빨리 학습하고 더 좋은 성능에 도달함을 의미. 보통 잠재공간의 차원(N)이 커질수록 모델이 복잡해져 과적합이 생기는데 변분 하한의 정규화(KL 발산) 성질 덕분에 불필요한 잠재 변수가 있어도 과적합되지 않는다고 설명.

Marginal Likelihood (주변 우도 비교)
원래 $p(x)$는 계산 불가능하지만, z 차원이 아주 낮으면(ex. 2~3차원) MCMC 추정으로 근사 값을 구할 수 있음. 이걸로 진짜 성능을 비교해보겠다는 뜻임. 주변 우도 추정기에 대한 자세한 정보는 부록에서 확인.
인코더와 디코더를 신경망으로 정의할 때, 100개의 은닉 유닛과 3개의 잠재 변수를 사용함. 더 높은 차원의 잠재 공간에서는 추정치가 신뢰할 수 없게 됨.

> Q. "더 높은 차원의 잠재 공간에서는 추정치가 신뢰할 수 없게 됨."의 의미가 무엇인가?
> A. VAE 모델 자체의 성능 문제가 아닌 성능을 측정하는 도구(MCEM)의 한계를 뜻함.
> VAE가 진짜로 좋은지 확인하기 위해 MCMC기반 추정기를 사용하여 진짜 주변 우도 $\log p(x)$ 값을 근사적으로 비교하려 함.
> 하지만 잠재 공간의 차원이 낮을 때 MCMC 추정기가 비교적 정확한 값을 내놓지 잠재 공간의 차원이 커질수록 차원의 저주 때문에 샘플링 효율이 급격히 떨어짐. 고차원에서 MCMC추정기가 내놓는 주변 우도 값의 오차가 너무 커져서 이 추정치를 기준으로 비교하는 것이 무의미해짐.

mnist 데이터셋이 사용되어 AEVB, WS은 하이브리드 몬테 카를로(HMC)[DKPR87] 샘플러를 사용하는 몬테 카를로 EM(MCEM)과 비교되었으며, 세부 사항은 부록에 있음. MCEM은 전통적인 통계학적 방법으로, 정확하지만 엄청 느림. 이걸 기준점으로 삼음.
작은 훈련 세트 크기(1,000)와, 큰 훈련 세트 크기(50,000)에 대해 세 알고리즘의 수렴 속도를 비교함. Figure 3: X축은 학습 진행도, Y축은 주변 로그 우도(값이 높을수록 좋음)
작은 데이터셋: MCEM(파란선)은 좋지만 느림. AEVB은 MCEM만큼 좋으면서 더 빠름.
큰 데이터셋: MCEM은 너무 느려서 수렴하지 못함. 반면 AEVB는 여전히 빠르고 높은 성능을 보임.

Visualisation of high-dimensional data(시각화)
잠재 공간을 시각화 가능한 차원 수(2,3)으로 설정한다면, 학습된 인코더를 사용하여 고차운 데이터를 저차원 매니폴드로 투영할 수 있음. mnist와 frey face 데이터셋에 대한 2D 잠재 매니폴드의 시각화는 부록 A(Figure 4)를 참조.

6. Conclusion

첫 번째 핵심 기여: SGVB 추정치를 통해 기존 방법들의 문제(intractability, high variance)를 해결. 이 추정치는 표준적인 SGD를 통해 간단하게 미분하고 최적화할 수 있음.
두 번째 핵심 기여: AEVB. 데이터 포인트마다 연속 잠재 변수가 있는 i.i.d 데이터셋의 경우(=사전 분포를 표준 가우시안으로 가정할 경우) SGVB를 통해 닫힌 해를 찾는 과정을 통해 추론과 학습을 효율적으로 해결함.

7. Future Work

SGVB 추정기와 AEVB 알고리즘은 연속 잠재 변수를 가진 거의 모든 추론 및 학습 문제에 적용될 수 있음.(범용성) 추후 많은 파생 모델이 나올 것임.
논문에서는 단순 MLP를 썼지만, CNN을 인코더/디코더로 쓰는 모델들이 나올 것을 예고. 잠재 변수 z를 한 층만 쓰는 게 아니라, $z_1 \rightarrow z_2 \rightarrow x$처럼 여러 층으로 쌓아서 더 복잡한 데이터를 표현하는 모델(계층적 VAE)
시간의 흐름이 있는 비디오, 음성 등과 결합한 Sequential VAE의 연구.
논문에서는 파라미터 θ, ϕ를 하나의 고정된 값으로 찾음(MAP). 하지만 SGVB 원리를 파라미터 자체에도 적용하면, 가중치조차도 확률 분포로 학습하는 완전한 베이지안 신경망을 만들 수 있음. [Appendix F]
데이터 x만 있는 비지도 학습뿐만 아니라, 레이블 y가 있는 지도 학습 환경에서도 잠재 변수 z를 활용할 수 있음.(Conditional VAE)

Appendix

Appendix A. Visualisations

figure 4: 2차원 잠재공간 Z과 그에 대응하는 관측 공간 X를 시각화함. 잠재 공간의 사전 분포가 가우시안이므로, 단위 정사각형 위의 선형적으로 간격을 둔 좌표들을 가우시간의 역 CDF(누적 분포 함수)를 통해 변환하여 잠재 변수 z의 값들을 생성함. 그냥 일정한 간격으로 자르는게 아닌, 백분위수를 활용하여 가우시안 분포의 중앙은 촘촘하게, 양 끝 부분은 넓게 좌표(z)가 잡힘. 이로써 데이터가 존재하는 영역을 더 균일하고 밀도 있게 시각화해줌. 이러한 각 z값에 대해, 학습된 파라미터 θ를 사용(디코더를 통해 매핑)하여 대응하는 생성 확률 $p_\theta(x|z)$(의 평균값, 분산 고려X)를 플롯함. 결과로 부드러운 변화가 나옴으로써 VAE가 단순히 이미지를 외운 것이 아닌, 데이터가 존재하는 연속적인 공간의 구조를 학습했다는 증거임.
figure 5: 다양한 차원 수의 잠재 공간에 대해, 학습된 mnist 생성 모델로부터 추출한 무작위 샘플

들. VAE의 잠재 공간의 차원이 높아져도 과적합 없이 안정적으로 학습되며, 더 많은 정보를 표현할 수 있음을 보임.

Appendix B. Solution of $-D_{KL}(q_\phi(z)||p_\theta(z))$, Gaussian case

변분 하한의 KL 발산 항은 종종 해석적으로 적분될 수 있는데, 이때 '해석적으로 적분된다'는 말은 복잡한 적분 기호를 풀어서 깔끔한 덧셈/뺄셈 공식으로 바꿀 수 있다는 뜻임.
이때 전제 조건은 사전 분포 $p(z)$는 표준 정규분포 $\mathcal{N}(0, I)$와 사후 근사 $q_\phi(z|x^{(i)})$가 모두 가우시안인 경우의 해를 제시.
J를 z의 차원 수라고 할때, μ, σ를 데이터 포인트 i에서 평가된 변분 평균 및 표준편차라 하고, μ_j, σ_j는 단순 이 벡터들의 j번째 요소를 나타냄.

$$-D_{KL} = -\int q(\log q - \log p)dz = \int q(\log p - \log q)dz = \int q \log p dz - \int q \log q dz$$

$\int q \log p$ 계산:

$$\int q_\theta(z) \log p(z) dz = \int \mathcal{N}(z; \mu, \sigma^2) \log \mathcal{N}(z; 0, I) dz = -\frac{J}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{J}(\mu_j^2 + \sigma_j^2)$$

$\log p(z)$는 표준 정규분포의 로그이므로 $-\frac{1}{2}z^2 -$ 상수 꼴임.
z^2의 기댓값($\mathbb{E}[z^2]$)은 분산 + 평균의 제곱($\sigma^2 + \mu^2$)임.
따라서 적분 결과는 $-\frac{1}{2}(\mu^2 + \sigma^2)$형태가 됨.

$\int q \log q$ 계산:

$$\int q_\theta(z) \log q_\theta(z) dz = \int \mathcal{N}(z; \mu, \sigma^2) \log \mathcal{N}(z; \mu, \sigma^2) dz = -\frac{J}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{J}(1 + \log\ \sigma_j^2)$$

가우시안 분포의 엔트로피(무질서도) 공식
가우시안의 엔트로피는 오직 분산(σ^2)에만 의존함.(평균 μ는 위치만 바꿀 뿐 무질서도에는 영향을 안줌.)
결과는 $-\frac{1}{2}(1 + \log\sigma^2)$형태가 됨.
$-D_{KL}((q_\phi(z)||p_\theta(z)) = \int q_\theta(z)(\log p_\theta(z) - \log q_\theta(z))dz = \frac{1}{2}\sum_{j=1}^{J}(1 + \log((\sigma_j)^2) - (\mu_j)^2 - (\sigma_j)^2)$
$-\frac{J}{2}\log(2\pi)$은 사라지고 결과적으로 $\frac{1}{2}\sum_{j=1}^{J}(1 + \log((\sigma_j)^2) - (\mu_j)^2 - (\sigma_j)^2)$만 남음.
도출된 식에는 적분($\int$)도 없고, 샘플링(z)도 없음. 인코더의 출력인 μ, σ만의 사칙연산으로 이루어져 있음.
이것은 실제 학습 시 KL loss를 구할 때 샘플링을 안 하고 `0.5 * sum(1 + log_var - mu^2 - exp(log_var))` 라는 코드를 쓰는가에 대한 수학적 증명임.

Appendix C. MLP's as probabilistic encoders and decoders

변분 오토인코더에서는 신경망이 확률적 인코더와 디코더로 사용됨. 데이터 유형과 모델에 따라 인코더와 디코더의 선택지는 다양해질 수 있음. 논문에서는 간단한 다층 퍼셉트론(MLP, Fully Connected Layer)를 사용. 인코더는 가우시안 출력을 가진 MLP를 사용했고, 디코더는 데이터 유형에 따라 가우시안 또는 베르누이 출력을 가진 MLP를 사용할 수 있음.

C.1 Bernoulli MLP as decoder
이 경우 $p_\theta(x|z)$를 다변량 베르누이 분포로 둠. 그 확률들은 단일 은닉층을 가진 완전 연결 신경망을 통해 z로부터 계산됨.
$\log p(x|z) = \sum_{i=1}^{D} x_i \log y_i + (1 - x_i) \cdot \log(1 - y_i)$
이는 BCE(Binary Cross Entropy) 손실 함수(의 음수)와 동일함.
y_i: 디코더가 예측한 i번째 픽셀이 1일 확률.
x_i: 실제 i번째 픽셀 값 (0 또는 1).
where $y = f_\sigma(W_2 \tanh(W_1 z + b_1) + b_2)$
입력: z
은닉층: $W_1 z + b_1 \rightarrow$ Tanh
출력층: $W_2(\text{hidden}) + b_2 \rightarrow$ Sigmoid (f_σ), (sigmoid를 쓰는 이유는 출력값 y가 확률(0~1)이어야 함.)

C.2 Gaussian MLP as encoder or decoder
인코더와 디코더를 대각 공분산 구조를 가진 다변량 가우시안으로 둠.
$\log p(x|z) = \log \mathcal{N}(x; \mu, \sigma^2 I)$ where $\mu = W_4 h + b_4$, $\log \sigma^2 = W_5 h + b_5$
여기서 평균 μ와 로그 분산 $\log \sigma^2$은 은닉층 h로부터 계산됨.
신경망이 분산(σ^2)을 직접 출력하지 않고 로그 분산($\log \sigma^2$)을 출력함. 분산은 항상 양수(+)여야 함. 신경망 출력에 제약이 없으면 음수가 나올 수 있음. 로그 분산을 출력하면, 나중에 지수 함수($\exp$)를 취했을 때 항상 양수가 되도록 보장할 수 있음.
$h = \tanh(W_3 z + b_3)$
은닉층 h까지는 μ와 σ가 경로를 공유함.
여기서 가중치와 편향들$\{W_3, W_4, W_5, b_3, b_4, b_5\}$은 MLP의 파라미터이며, 디코더로 사용될 때는 θ의 일부임.
인코더 구조도 위와 똑같음.
입력이 x, 출력이 z, 가중치와 편향은 변분 파라미터 ϕ가 됨. $\log q_\phi(z|x)$
은닉층: Tanh, 출력: μ_z, $\log \sigma_z^2$

Appendix D. Marginal Likelihood Estimator

주변 우도 $p(x)$를 계산하는 방법을 설명.
샘플링되는 공간의 차원 수가 낮고(5차원 미만) 충분한 샘플이 취해지는 한, 주변 우도에 대한 좋은 추정치를 산출하는 추정기를 유도함.
$p_\theta(x, z)$를 우리가 샘플링하고 있는 생성 모델이라고 하고, 주어진 데이터 포인트 $x^{(i)}$에 대해 주변 우도 $p_\theta(x^{(i)})$를 추정하고자 함. $p_\theta(x, z) = p_\theta(z)p_\theta(x|z)$. 이 추정 과정은 세 단계로 구성됨.

1단계: 사후 분포에서 샘플링.
진짜 사후 분포 $\log p_\theta(z|x)$는 모르지만 그 기울기$\nabla_z \log p_\theta(z|x)$는 우도와 사전 분포의 기울기 합:
$\nabla_z \log p_\theta(z|x) = \nabla_z \log p_\theta(z) + \nabla_z \log p_\theta(x|z)$
이 기울기 정보를 이용해서 HMC(Hybrid Monte Carlo)기법으로 사후 분포를 따르는 샘플 z들을 샘플링함.
2단계: 보조 밀도 함수 $q(z)$ 피팅.
위에서 샘플링한 샘플 $z^{(l)}$에 밀도 추정기 $q(z)$를 fit함.
뽑아낸 샘플들이 대략 어떤 모양의 분포를 이루는지, 가우시안 같은 쉬운 함수 $q(z)$로 감싸서 근사함.
3단계: 최종 추정. $p_\theta(x^{(i)})$를 바로 구할 수 없으니 역수를 취해 계산.
$p_\theta(x^{(i)}) \simeq (\frac{1}{L}\sum_{l=1}^{L} \frac{q(z^{(l)})}{p_\theta(z)p_\theta(x^{(i)}|z^{(l)})})^{-1}$ where $z^{(l)} \sim p_\theta(z|x^{(i)})$

$$\text{목표} : \frac{1}{p(x)}$$

$$\int q(z)dz = 1$$

$$\frac{1}{p_\theta(x^{(i)})} = \frac{1}{p_\theta(x^{(i)})} \cdot 1 = \frac{\int q(z)dz}{p_\theta(x^{(i)})}$$

여기서 $p_\theta(x^{(i)})$는 z와 상관없는 고정된 숫자(상수)임. 따라서 적분 기호 안으로 들어갈 수 있음.

$$= \int \frac{q(z)}{p_\theta(x^{(i)})}dz$$

이 식을 변형하기 위해 분모와 분자에 똑같은 수를 곱하는 트릭을 씀. 여기서 곱할 수는 $p_\theta(x^{(i)}, z)$ (결합 확률)임.

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

$$= \int \left(\frac{p_\theta(x^{(i)}, z)}{p_\theta(x^{(i)})}\right) \cdot \left(\frac{q(z)}{p_\theta(x^{(i)}, z)}\right)dz$$

앞쪽 괄호 $\frac{p_\theta(x^{(i)}, z)}{p_\theta(x^{(i)})}$는 베이즈 정리에 의해 사후 분포 $p_\theta(z|x^{(i)})$와 같음.

$$= \int p_\theta(z|x^{(i)}) \cdot \left(\frac{q(z)}{p_\theta(x^{(i)}, z)}\right)dz$$

적분의 정의: 어떤 확률 분포 p를 곱해서 적분한다는 것은 "그 분포 하에서의 평균(기댓값)을 구한다"는 뜻임. $\int p(z) \cdot f(z)dz = \mathbb{E}_{p(z)}[f(z)]$
적분의 정의에 따라 위 유도 수식이 진짜 사후 분포 $p_\theta(z|x^{(i)})$에서의 기댓값으로 표현됨.

$$= \mathbb{E}_{z \sim p_\theta(z|x^{(i)})}\left[\frac{q(z)}{p_\theta(x^{(i)}, z)}\right]$$

이 기댓값을 컴퓨터가 계산할 수 있는 덧셈으로 바꿈.

$$\approx \frac{1}{L}\sum_{l=1}^{L}\frac{q(z^{(l)})}{p_\theta(x^{(i)},z^{(l)})}$$

이때 $p_\theta(x,z)=p_\theta(x|z)p_\theta(z)$이므로 분모는 계산 가능함.
위 수식의 의미가 진짜 사후 분포$p(z|x)$에서 뽑은 샘플 $z\sim p_\theta(z|x^{(i)})$들 가지고, $\frac{q(z)}{p(x,z)}$라는 비율을 계산해서 평균을 내면 그게 목표였던 $\frac{1}{p(x)}$임.
부록 D 요약:

- 구하려는 값의 역수$(1/p)$를 적분식으로 씀.
- 분자 분모에 $p(x,z)$를 곱해서 베이즈 정리$(p(z|x))$ 모양을 만듦.
- 그러면 식이 "$p(z|x)$에서 뽑은 샘플들로 $\frac{q}{p(x,z)}$의 평균을 구하라"는 뜻이 됨.
- 평균을 구한 뒤, 다시 역수를 취해서 원래 값(p)을 찾음.

Appendix E. Monte Carlo EM

몬테 카를로 EM알고리즘은 인코더를 사용하지 않음. 매번 데이터 x가 주어질 때마다 z공간을 탐색하며 어디가 확률이 높은 z인가를 찾아야 함. 대신 사후 분포의 기울기$\nabla_z \log p_\theta(z|x)$를 사전 분포의 기울기와 우도의 기울기의 합 $=\nabla_z \log p_\theta(z)+\nabla_z \log p_\theta(x|z)$ 으로 계산할 수 있음.
이 몬테 카를로 EM절차는 z값 하나를 제대로 찾기 위해, HMC(Hybrid Monte Carlo)라는 복잡한 시뮬레이션을 10번이나 수행해야 함.(매우 느린 이유) 그렇게 힘들게 z를 찾은 뒤에야, 비로소 파라미터 θ를 5번 업데이트함. 학습보다 탐색(z 찾기)에 더 많은 비용을 쓰고 있음.
모든 알고리즘(VAE, WS, MCEM)에 대해 파라미터를 최적화하는 옵티마이저는 Adagrad(어닐링 스케줄이 동반된)로 통일함.
주변 우도(성능 평가 지표)는 훈련 및 테스트 세트의 첫 1,000개 데이터 포인트로 추정되었으며, 각 데이터 포인트마다 4번의 립프로그 단계를 가진 HMC를 사용하여 잠재 변수 사후 분포에서 50개의 값을 샘플링함.

> Q. 립프로그 단계란 무엇인가?
> A. 하이브리드 몬테 카를로 샘플링 방법에서 입자의 움직임을 시뮬레이션하기 위해 사용하는 수치 적분 방법임.
> 자세한건... 나중에 다시 보도록 하자.

Appendex F. Full VB

파라미터 θ를 고정된 값으로 취급한 방법의 확장으로, 파라미터까지도 확률 변수(분포)로 취급하여 학습하는 방법을 다룸. 이것이 진정한 의미의 베이지안 신경망 접근 방식임.
본문의 방법(AEVB): 인코더(ϕ)와 디코더(θ)의 가중치를 하나의 최적값으로 찾음.(MAP 추정)

Full VB 방법: 가중치 θ 자체도 불확실성을 가진 확률 분포(ex. 가우시안)로 정의하고, 그 분포의 파라미터(평균, 분산)를 학습함. 이렇게 하면 모델의 불확실성을 더 잘 다룰 수 있음. 부록 F에서 이 경우에 대한 추정치를 유도할것임.
α에 의해 매개변수화된, 어떤 초사전 분포(hyperprior)를 $p_\alpha(\theta)$라고 정의함. 파라미터 θ도 확률 변수이므로, 학습 전 사전 분포(ex. 평균 0인 가우시안)를 가정해야 함.
이때 주변 우도를 정의하면 $\log p_\alpha(X) = D_{KL}(q_\phi(\theta)||p_\alpha(\theta|X)) + \mathcal{L}(\phi; X)$임.
본문의 식과 형태가 똑같지만 대상이 잠재 변수 z에서 파라미터 θ로 바뀜. 이제 목표는 $\log p_\alpha(X)$(데이터의 총 확률)를 최대화하는 것이며, 이를 위해 하한인 $\mathcal{L}(\phi; X)$를 최대화해야함.
우변의 첫번째 항은 근사 사후 분포($q_\phi(\theta)$)와 실제 사후 분포($p_\alpha(\theta|X)$, intractable) 사이의 KL 발산을 나타내며, $\mathcal{L}(\phi; X)$는 주변 우도에 대한 변분 하한을 나타냄.(KL 발산은 non-negative이므로 $\mathcal{L}(\phi; X)$가 주변 우도에 대한 하한이 됨. 근사 사후 분포와 실제 사후 분포가 정확히 일치할 때 이 하한은 실제 주변 우도와 같아짐.)
$\mathcal{L}(\phi; X) = \int q_\phi(\theta)(\log p_\theta(X) + \log p_\alpha(\theta) - \log q_\phi(\theta))d\theta$
이 ELBO를 계산하려면 모든 가능한 파라미터 θ에 대해 적분을 해야 함.
$\log p_\theta(X)$은 개별 데이터 포인트들의 주변 우도의 합 $= \sum_{i=1}^{N} \log p_\theta(x^{(i)})$으로 구성됨.
$\log p_\theta(X) = \sum_{i=1}^{N} \log p_\theta(x^{(i)})$
이는 아래와 같이 다시 표현됨.
$\log p_\theta(x^{(i)}) = D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z|x^{(i)})) + \mathcal{L}(\theta, \phi; x^{(i)})$
여기서 중요한 점은 $\mathcal{L}(\phi; X)$안에 $\log p_\theta(X)$가 들어있고, 그 $\log p_\theta(X)$ 안에는 다시 식 (15)의 $\mathcal{L}(\theta, \phi; x)$가 들어있다는 계층 구조임. 즉, 이중 적분 구조가 됨.(바깥쪽은 θ적분, 안쪽은 z적분)
여기서 다시 우변의 첫번째 항은 근사 사후 분포와 실제 사후 분포 사이의 KL 발산이며, $\mathcal{L}(\theta, \phi; x)$는 데이터 포인트 i의 주변 우도에 대한 변분 하한임.
$\mathcal{L}(\theta, \phi; x^{(i)}) = \int q_\phi(z|x)(\log p_\theta(x^{(i)}|z) + \log p_\theta(z) - \log q_\phi(z|x))dz$
$\mathcal{L}(\phi; X)$와 $\mathcal{L}(\theta, \phi; x^{(i)})$의 우변에 대한 기대치는 분명 세개의 개별 기대값의 합으로 작성될 수 있으며, 두 번째와 세 번째 구성 요소는 때때로 해석적 해를 구할 수 있음. 예를 들어 $p_\theta(x), q_\phi(z|x)$가 모두 가우시안일 때에 해당하는데, 일반적으로 이러한 기대가 각각 다루기 어렵다고 가정함.
재매개변수화 트릭의 이중 적용
z뿐만 아니라 θ에도 재매개변수화 트릭을 씀.

Standard VAE는 z를 확률 변수로 취급하여 $q_\phi(z|x)$를 추론하지만, 신경망의 weight θ는 고정된 값을 찾는 최적화 대상임. Full VB에서는 θ도 불확실성을 가진 확률 변수로 보고, 이에 대한 사후 분포 $p(\theta|X)$를 근사하는 $q_\phi(\theta)$를 학습함. 이를 통해 모델 파라미터 자체의 불확실성을 모델링할 수 있음. 이 Full VB를 위한 추정량을 유도함.
θ가 확률 변수가 되었으므로, 이에 대한 사전 분포(hyperprior)가 필요. 이를 $p_\alpha(\theta)$로 정의. α로 매개변수화 된다고 가정함.
데이터 셋 X에 대한 주변 로그 우도는 다음과 같이 표현됨.
$\log p_\alpha(X) = D_{KL}(q_\phi(\theta)||p_\alpha(\theta|X)) + \mathcal{L}(\phi; X)$
좌변: 데이터의 로그 우도.(Marginal Log Likelihood)
우변 첫째 항: 근사 사후 분포 $q_\phi(\theta)$와 실제 사후 분포 $p_\alpha(\theta|X)$간의 KL divergence(항상 0보다

큼.)
우변 둘째 항: Marginal Likelihood의 변분 하한(ELBO).
ELBO 항($\mathcal{L}$)을 최대화하는 것이 곧 $\log p(X)$를 최대화하거나 KL을 최소화하는 것과 같음.

> Q. ELBO를 최대화하는 것이 KL을 최소화하는 것과 왜 같은가?
> A. 변분 추론의 항등식 때문. 구하고자 하는것은 데이터 x가 주어졌을 때 모델의 우도, $\log p(x)$를 최대화하는 것인데, 이 로그 우도가 다음과 같이 분해됨.
> $\log p(x) = \underbrace{\mathcal{L}(\phi; x)}_{\text{ELBO}} + \underbrace{D_{KL}(q_\phi(z|x)||p(z|x))}_{\text{KL Divergence}}$
> 이때 데이터 x는 고정되어 있으므로, 모델의 로그 우도는 변분 파라미터 ϕ와는 무관한 고정된 상수임. 따라서 우변의 합도 일정해야 하므로 ELBO가 증가하면 KL 발산은 감소해야 함. 따라서 ELBO를 최대화한다는 것은, 근사 분포를 실제 사후 분포에 최대한 가깝게 만들어 오차를 줄이는 것과 수학적으로 동치임.

위 식의 ELBO항을 다음과 같이 정의할 수 있음.
$\mathcal{L}(\phi; X) = \int q_\phi(\theta)(\log p_\theta(X) + \log p_\alpha(\theta) - \log q_\phi(\theta))d\theta$
일반 VAE의 ELBO와 다르게, 바깥쪽에 θ에 대한 적분$\int q_\phi(\theta)d\theta$가 추가됨.
즉, 파라미터 θ가 $q_\phi(\theta)$를 따를 때의 기대값을 구하는 형태가 됨.
KL발산은 non-negative이므로 ELBO항이 데이터 우도의 하한이 됨(하한 자체를 최대화하면 우도도 같이 높아지는 것). 근사 사후분포와 실제 사후분포가 정확히 일치할 때 이 하한은 실제 marginal likelihood와 같아짐.
$\log p_\theta(X)$항은 개별 데이터 포인트들의 주변 우도의 합 $\sum_{i=1}^{N} \log p_\theta(x^{(i)})$으로 구성되며, 이를 다음과 같이 쓸 수 있음.
(각 데이터 포인트에 대한 로그 우도 분해 식)
$\log p_\theta(x^{(i)}) = D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z|x^{(i)})) + \mathcal{L}(\theta, \phi; x^{(i)})$
standard VAE의 식과 같음. 한가지 다른 건 위 식 전체가 θ에 대한 기댓값($\mathcal{L}(\phi; X)$) 안에 포함됨.
이때 우변의 첫 번째 항은 실제 사후분포(z에 대한)와 근사 분포 간의 KL 발산이며, $\mathcal{L}(\theta, \phi; x^{(i)})$는 데이터 포인트 i의 주변 우도에 대한 변분 하한임.
(데이터 포인트 i에 대한 변분 하한 정의)
$\mathcal{L}(\theta, \phi; x^{(i)}) = \int q_\phi(z|x)(\log p_\theta(x^{(i)}|z) + \log p_\theta(z) - \log q_\phi(z|x))dz$
위 식이 잠재 변수 z에 대해 적분한 형태인 Standard VAE의 ELBO임.
두 ELBO의 우변에 있는 기댓값들은 세 개의 개별적인 기댓값의 합으로 쓸 수 있으며, 그중 두 번째와 세 번째 요소(KL 항)는 p와 q가 모두 가우시안일 경우처럼 해석적으로 풀 수 있음. 그치만 일반성을 위해 여기서 이 기댓값들이 모두 intractable하다고 가정할 것임.
본문과 동일하게 두 근사 사후 분포 $q_\phi(\theta)$, $q_\phi(z|x)$에 대해 $\tilde{z} \sim q_\phi(z|x)$를 보조 노이즈 변수 ϵ을 사용하여 $\tilde{z} = g_\phi(\epsilon, x)$로 재매개변수화할 수 있음
이때 z에 대한 기댓값을 ϵ에 대한 기댓값으로 바꿀 수 있음.
$\mathcal{L}(\theta, \phi; x^{(i)}) = \int q_\phi(z|x)(\log p_\theta(x^{(i)}|z) + \log p_\theta(z) - \log q_\phi(z|x))dz$
$= \int p(\epsilon)(\log p_\theta(x^{(i)}|z) + \log p_\theta(z) - \log q_\phi(z|x))|_{z=g_\phi(\epsilon, x^{(i)})}d\epsilon$
$q_\phi(\theta)$에 대해서도 동일하게 적용할 수 있음. $\tilde{\theta} = h_\phi(\zeta)$ with $\zeta \sim p(\zeta)$. ζ가 위 식의 ϵ임. 이처럼 파라미터 θ도 확률 변수이므로, 이를 직접 샘플링하응 대신 고정된 노이즈 분포 $p(\zeta)$(hyperprior, 보통

표준가우시안)에서 ζ를 뽑고, 미분 가능한 함수 h_ϕ를 통해 θ를 생성함. 이로써 θ의 파라미터(ex. weight의 평균과 분산)에 대해 역전파가 가능함.
$\mathcal{L}(\phi;X) = \int p(\zeta)(\log p_\theta(X) + \log p_\alpha(\theta) - \log q_\phi(\theta))|_{\theta=h_\phi(\zeta)} d\zeta$
이제 전체 데이터셋에 대한 하한 $\mathcal{L}(\phi;X)$는 θ에 대한 적분이 아닌 ζ에 대한 적분 형태로 표현됨.
따라서 최종 추정량(The SGVB Estimator for Full VB)은 다음과 같이 표현됨.
$f_\phi(x,z,\theta) = N \cdot (\log p_\theta(x|z) + \log p_\theta(z) - \log q_\phi(z|x)) + \log p_\alpha(\theta) - \log q_\phi(\theta)$
$N \cdot (\ldots)$: 미니배치 처리를 위해 전체 데이터 개수 N을 곱하여 스케일을 맞춘 부분임. 괄호 안은 데이터 포인트 하나($x^{(i)}$)의 재구성 오차와 z의 KL항 관련 부분임.
뒷부분: θ에 대한 prior와 근사 사후분포 항임.
재매개변수화된 두 ELBO를 사용하여 데이터 포인트 $x^{(i)}$가 주어졌을 때 변분 하한의 몬테카를로 추정값은 다음과 같음.
$\mathcal{L}(\phi;X) \simeq \frac{1}{L}\sum_{l=1}^{L} f_\phi(x^{(l)}, g_\phi(\epsilon^{(l)}, x^{(l)}), h_\phi(\zeta^{(l)}))$
z를 위한 노이즈 ϵ과, θ를 위한 노이즈 ζ를 모두 샘플링.
$z = g(\epsilon, x)$와 $\theta = h(\zeta)$를 계산.
이를 함수 f에 대입하여 평균을 냄.
여기서 $\epsilon^{(l)} \sim p(\epsilon)$이고 $\zeta^{(l)} \sim p(\zeta)$임. 이 추정량은 ϕ의 영향을 받지 않는 $p(\epsilon)$과 $p(\zeta)$에서의 샘플에만 의존하므로, ϕ에 대해 미분 가능함.

> Q. 그럼 신경망의 파라미터(최적화 대상)는 ϕ밖에 남지 않은것인가? 디코더의 θ만 확률 변수로 생각되어지는 것인가?
> A. Standard VAE는 ϕ, θ 가 각각 인코더, 디코더 신경망의 weight(고정값)이였음.
> Full VB VAE에서는 θ 자체가 확률 변수가 됨. 즉, 웨이트가 하나의 숫자가 아닌 분포를 가짐. ϕ는 이 θ의 분포(근사 사후분포 $q_\phi(\theta)$)를 결정하는 결정론적 파라미터(weight의 평균과 분산)가 됨. 따라사 경사하강법으로 업데이트해야 하는 유일한 최적화 변수는 ϕ가 맞음.
> 논문 부록 F 표기를 보면 생성 모델의 파라미터 θ라고 통칭하고 이를 확률 변수로 취급함. 일반적으로 VAE에서 θ는 디코더의 파라미터를 지칭하는데, 따라서 문맥상 디코더의 웨이트를 확률 변수로 취급함. 하지만 Full VB의 철학을 확장하면 인코더의 웨이트ϕ 또한 확률 변수로 취급할 수 있음. 다만, 부록 F 표기에서는 생성 모델(디코더) 파라미터 θ에 대한 베이지안 추론을 강조함. Full VB 설정에서 신경망의 '웨이트'값들(θ)은 매번 샘플링되는 확률 변수 취급을 받으며, 우리가 학습(업데이트)하는 것은 그 웨이트의 분포를 정의하는 평균과 분산 값들(ϕ)임.

F.1 Example 가우시안 예제

파라미터 θ와 잠재 변수 z에 대한 prior을 중심이 0인 등방성 가우시안 $p_\alpha(\theta) = \mathcal{N}(\theta;0,I)$ and $p_\theta(z) = \mathcal{N}(z;0,I)$이라고 한다면 두 변분 근사 사후 분포를 대각 공분산 구조를 가진 다변량 가우시안으로 설정할 수 있음.
$\log q_\phi(\theta) = \log \mathcal{N}(\theta;\mu_\theta,\sigma_\theta^2 I)$
$\log q_\phi(z|x) = \log \mathcal{N}(z;\mu_z,\sigma_z^2 I)$
z에 대한 분포는 인코더 네트워크의 출력(μ_z, σ_z)으로 정의됨.
θ에 대한 분포는 이제 학습 가능한 파라미터 $\mu_\theta, \sigma_\theta$를 가짐. 즉, 모든 weight가 하나의 값이 아닌 평

균과 분산을 가짐.
이때 μ_z, σ_z는 x에 대한 함수(인코더)임.

Algorithm 2

Algorithm 2 Pseudocode for computing a stochastic gradient using our estimator. See text for meaning of the functions $f_{\boldsymbol{\phi}}$, $g_{\boldsymbol{\phi}}$ and $h_{\boldsymbol{\phi}}$.

Require: $\boldsymbol{\phi}$ (Current value of variational parameters)
  $\mathbf{g} \leftarrow 0$
  for l is 1 to L **do**
    $\mathbf{x} \leftarrow$ Random draw from dataset $\mathbf{X}$
    $\boldsymbol{\epsilon} \leftarrow$ Random draw from prior $p(\boldsymbol{\epsilon})$
    $\boldsymbol{\zeta} \leftarrow$ Random draw from prior $p(\boldsymbol{\zeta})$
    $\mathbf{g} \leftarrow \mathbf{g} + \frac{1}{L} \nabla_{\boldsymbol{\phi}} f_{\boldsymbol{\phi}}(\mathbf{x}, g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x}), h_{\boldsymbol{\phi}}(\boldsymbol{\zeta}))$
  end for
  return $\mathbf{g}$

일반적인 VAE(Algorithm 1)가 잠재 변수 z만 샘플링하는 것과 달리, 모델의 파라미터 θ까지 노이즈 ζ를 통해 샘플링한다는 점임.
Require: 변분 파라미터의 현재 값 ϕ.(이때 ϕ는 인코더의 가중치뿐만 아니라, θ의 분포(평균, 분산)를 결정하는 파라미터도 포함.)
Init: $g \leftarrow 0$ (기울기 누적 변수 초기화)
Loop: for l = 1 to L do (샘플링 횟수 L만큼 반복)
1. Data: $x^{(l)} \leftarrow$ 데이터셋 X에서 무작위 추출 (미니배치 혹은 하나의 데이터를 뽑음.)
2. Noise (z): $\epsilon^{(l)} \sim p(\epsilon)$ (잠재 변수 z를 생성하기 위한 노이즈 추출)
3. Noise (θ): $\zeta^{(l)} \sim p(\zeta)$ (파라미터 θ를 생성하기 위한 노이즈 추출)
4. Calc: $g \leftarrow g + f_\phi(x^{(l)}, g_\phi(\epsilon^{(l)}, x^{(l)}), h_\phi(\zeta^{(l)}))$ (추정된 목적 함수의 값을 누적)
end
return g (계산된 추정값 반환 -> 이후 미분하여 파라미터 업데이트)
이 Algorithm 2는 베이지안 신경망을 학습시키는 방식과 동일함.

또한, 두 확률 변수의 prior이 가우시안이므로, 다음과 같이 근사 사후 분포를 매개변수화할 수 있음.
$q_\phi(\theta)$ as $\tilde{\theta} = \mu_\theta + \sigma_\theta \odot \zeta$, where $\zeta \sim \mathcal{N}(0, I)$
$q_\phi(z|x)$ as $\tilde{z} = \mu_z + \sigma_z \odot \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$
여기서 $\odot$는 요소곱, 위 두 근사 사후 분포는 위에 정의한 두 ELBO와 연결지을 수 있음.

이때 모델 내 4개 항을 해석적으로 풀 수 있으므로 분산이 더 낮은 대체 추정량을 구성할 수 있음.
(부록 F 앞쪽에서 '일반성을 위해 intractable하다고 가정'했지만, 가우시안의 경우 KL 발산 항을 적분 없이 바로 수식으로 해석적 해를 찾을 수 있음.(부록 B참조.) 이렇게 하면 샘플링 노이즈가 줄어들어 학습이 안정됨.)

$$\mathcal{L} \simeq \frac{1}{L} \sum_{l=1}^{L} \left(N \cdot (\mathrm{KL}(z \text{ terms}) + \log p(x|z)) + \mathrm{KL}(\theta \text{ terms}) \right)$$

위 식은 z에 대한 KL 항(해석적 계산), θ에 대한 KL 항(해석적 계산), 재구성 오차(샘플링 필요)의 합으로 구성됨.
이렇게 하면 Bayesian Neural Network와 유사하게 가중치에 대한 정규화 효과(KL 항이 가중치가 Prior인 0 근처에 머물도록 유도)를 얻으면서 동시에 생성 모델이 학습할 수 있음.

> Q. 베이지안 신경망(Bayesian Neural Network, BNN)이란?
> A. 기존 신경망과 달리 가중치가 고정된 값이 아니라 확률 분포를 따르는 신경망을 의미.
> 일반 신경망 (Deterministic NN):
> W = 0.5(하나의 고정된 숫자)
> 학습 결과: 최적의 숫자 하나를 찾음.
> 예측: 항상 같은 입력에 같은 출력.
> 베이지안 신경망(BNN):
> $W \sim \mathcal{N}(0.5, 0.01)$ (가중치는 평균과 분산을 가지는 분포에서 샘플링.)
> 학습 결과: 가중치의 최적 분포(평균과 분산)를 찾음.
> 예측: 예측할 때마다 가중치 분포에서 값을 샘플링하여 계산하므로, 결과가 확률적으로 변동함.
> 논문의 부록 F가 정확히 VAE의 디코더를 베이지안 신경망으로 만드는 과정임.
> Algorithm 2에서 Noise (ζ)를 뽑아 θ를 생성하는 과정이 BNN의 순전파임.
> Q. 그럼 왜 베이지안 신경망을 사용하는가?
> A. 1. 불확실성 추정: 모델이 자신의 예측을 얼마나 확신하는지 알 수 있음.(분산이 크면 불확실함). 2. 정규화 효과(Regularization): KL divergence 항이 가중치가 보통 prior(보통 0)에서 너무 멀어지지 않도록 강제하므로, 과적합을 매우 강력하게 방지함. 데이터가 적을 때 특히 유리함.

구현 관점에서 보면 `nn.Linear` 레이어의 weight가 고정된 `Parameter`가 아닌, μ와 σ를 파라미터로 가지고 `forward()` 때마다 ϵ을 뽑아 $w = \mu + \sigma \cdot \zeta$를 계산하는 베이지안 신경망으로 대체되어야 함.