Tutorial on VAE

Yongjin Shin

November 26, 2018

1. Introduction

"Generative Model"은 주어진 데이터 x의 분포 $p(x)$를 다루는 아주 넓은 분야를 말한다. 여기서 x는 고차원 공간이 X에 존재하는 하나의 데이터라 할 수 있겠다. 이러한 모델을 훈련하기 위해서는 몇 가지 문제점들이 있어왔다.

(1) 우리는 데이터의 분포에 대해서 강력한 가정을 했어야만 했다.

(2) MCMC와 같이 계산 시간과 비용이 많이 든다.

그러나 backpropagation을 통해서 아주 강력한 approximator를 생성이 가능한 딥러닝 모델에 대한 연구가 활발해지면 이러한 Generative Model에 대한 해결책이 등장하기 시작했다. Variational Autoencoder(VAE)는 그 방법론 중에 하나인데, 앞에서 문제시했던 데이터 분포에 대한 사전 가정은 줄어들고 backpropagation을 통해 계산시간을 획기적으로 줄일 수 있게 되었다. 뿐만 아니라 비록 VAE가 오차를 발생시키기는 하지만, 공간 X의 차원의 크기를 고려한다면 충분히 작은 오차를 생성한다고 할 수 있겠다.

1.1 Variational Inference

Generative Model을 학습할 때 공간 X에서 각 차원 별로 연관(dependency)이 깊을수록 그리고 더욱 복잡한 구조일수록 학습은 어려워진다. 따라서 우리는 잠재 변수(Latent Variable, Hidden Variable)을 도입하게 된다. 예컨대 MNIST에서 각각의 관측값(Observed Variable)을 보면 단순히 Label이 1, 2, 3, .. 이라는 것 이외에 획의 굵기라든지 숫자의 기울기와 같은 특징들을 잡아낼 수 있다. 이렇게 관측값들을 결정하는 higher-level의 변수를 잠재변수라고 할 수 있겠다. MNIST 데이터는 우리가 눈으로 보아도 쉽게 파악할 수 있는 구조이기 때문에 (단지 몇 개의) 잠재변수를 확인할 수 있다. 그러나 고차원 공간의 데이터의 경우에는 우리가 잠재변수를 가늠할 수가 없을 뿐더러 잠재변수 역시 상상이상으로 고차원 공간 Z일 수 있다. Generative Model에서는 차원 사이의 연관성에 관여하는 잠재변수 z 의 분포 $p(z)$를 파악해내고 이를 통해 다시 공간 X로 예측값 x를 생성하는 것이 중요하다고 할 수 있겠다. 그렇다면 우리는 잠재변수를 어떻게 확인할 수 있을까? 우리의 모델 $p(X, Z)$를 떠올려보자. Bayes Theory에 의해 다음과 같은 수식을 얻을 수 있다.

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} \tag{1}$$

우리가 원하는 잠재변수에 대한 힌트를 얻기 위해서는 $p(x)$를 알아내야한다. 그런데 $p(x)$를 알기 위해서는 결국 *식(2)*와 같이 잠재변수에 대해서 marginalize를 해야만 한다. 결국, 모르는 것을 알아내기 위해서 모르는 것을 사용해야한다는 문제에 봉착하게 된다. 따라서 이 문제를 해결하기 위해서 제안되는 방법들이 Variational Inference와 MCMC라 할 수 있다. 그 중 Variational Inference의 중요한 컨셉은 우리가 알고자 하는 분포 p가 아닌 **임의의** 분포 q를 차용하여 $p(X)$를 근사하겠다는 것이다. (더 이상 자세한 설명은 생략한다. Appendix를 참고할 것.) 그러나 이 방법들은 앞에서 언급한 문제를 각각 지니고 있기 때문에 한계를 가진다고 할 수 있다.

2. Variational Autoencoder

2.1 An implicit Generative Model (a.k.a Decoder

다시 더 큰 틀에서 다시 우리의 모델을 바라보도록 하자. 우리가 원하는 모델은 분포 $p(X)$를 알아내서 주어진 데이터 $x \in X$들과 최대한 비슷한 예측값들을 생성해내는 것이다. 이것을 위해서 잠재변수 $z \in Z$를 사용할 수 있겠다. 즉, 이를 Bayes Theory에 적용해서 풀어내자면 다음과 같은 수식으로 풀어낼 수 있다.

$$p(x) = \int_z p(x|z;\theta)p(z)dz \tag{2}$$

여기서 x는 공간 X에 있는 한 데이터 포인트, z는 공간 Z에 있는 잠재변수, 그리고 θ는 Θ라는 연속인 공간에 있는 변수라고 하자. $p(x|z;\theta)$는 z, θ가 정해졌을 때 생겨나는 (x를 domain으로 하는) Probability Density Function(PDF)라고 할 수 있다. 따라서 위의 식은 특정 θ가 주어지고, 모든 잠재변수 z에 대해서 각각 결정되는 PDF을 모아놓은 것이 $p(x)$라고 말한다. VAE는 $p(x|z;\theta)$ (likelihood)를 다음과 같이 가정한다 (가정이 강력하지 않을 뿐 가정이 없는 것은 아니다. VAE에서는 대략 3,4개의 가정이 존재한다).

$$p(x|z;\theta) = \mathcal{N}(x|f(z;\theta), \sigma^2 I) \tag{3}$$

즉, Likelihood를 Gaussian Distribution으로 가정하였다. 그런데 Gaussian Distribution의 두 변수인 평균과 표준편차는 $f(z;\theta)$와 $\sigma^2 I$ 두었다. 여기서 σ는 hyper parameter로 이 분포는 isometric이라는 것을 알았다. 그렇다면 $f(z;\theta)$는 뭘까? 위에서 밝혔듯이 $p(x|z;\theta)$는 z와 θ에 의해 바뀐다고 하였는데, 어떻게 바뀌느냐가 평균이 $f(z;\theta)$에 의해 결정되면서 Gaussian이 이동한다고 상상을 하면된다.

그럼 우리가 모르는 변수 z, θ 가 있다. 우선 z에 대해서는 어떻게 해야만 할까? 일단 Z를 어떻게 정의(예를들어 첫번째 차원은 획의 굵기, 두번째 차원은 글꼴의 기울기)할 수 있을까? VAE에서 Z는 Multivariate Normal Distribution $\mathcal{N}(0, I)$을 따른다고 가정한다. 아니, Gaussian Distribution으로 복잡한 데이터를 유추한다는게 말이 되는가?라고 반문 할 수 있다. 그 해결방법은 Inverse Transform Sampling이라는 것을 통해 이루어진다. 어떤 function f를 떠올려보자. 고등학교 때 f^{-1}를 계산해서 codomain(y)을 통해서 domain(x)를 구하는 문제를 무수히도 많이 풀었다. 똑같이 우리는 $p(X)$에서 sampling을 통해서 얻은 x를 주어진 f^{-1}에 통과시켜 $p(Z) \sim \mathcal{N}(0, I)$를 구하면 된다. 반대로 말하면 function $f(z;\theta)$만 적절하게 주어진면 $\mathcal{N}(0, I)$를 통해서 주어진 데이터의 분포와 동일한 분포를 만들어낼 수 있다. 아래 그림은 uniform distribution(codomain)에서 샘플링을 하여 특정 f인 exponential distribution의 CDF를 통과시킨 모습이다.

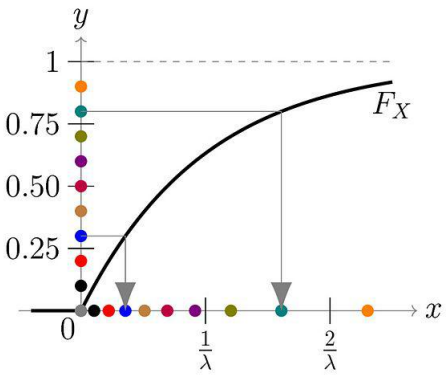

Figure 1

따라서 새로운 이슈는 parameter θ와 z를 domain으로 가지는 deterministic function $f(z;\theta)$를 어떻게 구하느냐 이다. 다행히도 뉴럴넷이 하는 일이 function $f(z;\theta)$를 만드는 일이다. 그럼 θ (고정된 값. x, z와 달리 확률 변수가 아님)는 어떻게 처리해야할까? 어떤 θ 를 사용해야만 실제 데이터 분포 $p(x)$를 근사하는 함수를 만들 수 있을까? 이는 곧 Maximum Likelihood Estimation 문제가 된다. 그러나 Gaussian Distribution은 미분이 가능하고 θ는 연속한 공간 상에 있으므로 Gradient Descent를 사용해서 최적점을 찾아낼 수 있다. 뉴럴넷에서는 backpropagation을 통해 최적화가 이루어진다.
(조그만 생각해보면 왜 이 과정이 decoder라 불리는지 알 수 있다.)

정리를 하자면, 우리는 특정 $f(z;\theta)$라는 함수만 잘 구할 수 있으면 Normal distribution에서 샘플링을 해서 우리가 원하는 $p(X)$와 유사한 분포를 만들어 낼 수 있다. 그리고 최적의 θ는 뉴럴넷의 backpropagation을 통해서 구할 수 있게 된다. 이 모든 과정은 Encoder에서 전달받은 Encoded data($p(z)$)를 어떻게 잘 풀어낼 수 있는지와 관련된 Decoding 작업이라고 할 수 있겠다.

2. Variational Inference for the Posterior (a.k.a Encoder)

z를 정의했고, θ는 backpropagation을 통해 최적점을 찾으면 된다. 그럼 우리는 $f(z;\theta)$를 구할 수 있게 된다. 그런데 다시 식(2)로 돌아가보자. 가만보면, 적분을 해야하는데 어느 세월이 이걸 다 계산할 수 있을까? 만약 공간 Z가 엄청나게 고차원에 있다면, 우리가 샘플링할 수 있는 $z \in Z$는 엄청난 수로 증가해버린다. 결론만 우선 말하자면 VAE는 주어진 데이터 x들을 통해서 엄청나게 높은 차원의 Z에서 쓸모있을만한

것들만 걸러내서 차원을 줄여버린다 ($Q(Z|X)$, 여기서 Q를 왜 사용했을까? 잘 모르겠다면 Appendix를 다시 읽어보자). 그러면 훨씬 적은 계산으로 우리의 목적을 달성할 수 있다. 물론 계산시간을 줄이기 위해 approximation을 했고, 이 때문에 오차가 생길 수 밖에는 없다.

우리가 임의의 분포 $Q(Z|X)$를 도입하는 순간 Decoder에서 가정한 $p(Z) \sim \mathcal{N}(0, I)$는 더 이상 유효하지 않는 것처럼 보인다. 왜냐하면 $p(Z)$에서는 원래 고차원이었던 Z에 대한 분포이고,$Q(Z|X)$는 사실 줄어든 차원 Z'에 대한 분포를 말하기 때문이다. 물론 앞에서도 언급했듯이 $Q(Z'|X)$는 충분히 데이터를 잘 설명할 수 있는 잠재변수를 지니고 있다. 그럼에도 불구하고 $p(Z)$와 $Q(Z'|X)$는 차원도 다르고 분포도 다르다. 하지만 이 사실은 (우리가 제일 잘하는 어려운건 나중에) 미뤄두도록 하고 , $Q(Z'|X)$가 어떻게 데이터를 잘 설명할 수 있는지를 먼저 생각해보자. ($Q(Z'|X)$는 $Q(Z|X)$로 표기할 것이니 혼동하지 말 것. 그리고 복잡해 보이는 수학 공식 출현 주의.)

자, 앞에서 언급한 이야기들을 수학적으로 표현을 하자면 아래와 같다.

$$\begin{aligned} KL[Q(z|X)||p(z|X)] &= \mathbb{E}_{z\sim Q}[\ln Q(z|X) - \ln p(z|X)] \\ &= \mathbb{E}_{z\sim Q}[\ln Q(z|X) - \ln p(X|z) - \ln p(z)] + \ln p(X) \end{aligned} \tag{4}$$

위의 식에서 두번째 등호에서 $\ln p(x)$는 z와 전혀 관계가 없으므로 $\mathbb{E}$ 밖으로 튀어나오게 된다. $\ln p(x)$를 왼쪽으로 넘겨보면 아래와 같이 표현할 수 있다.

$$\begin{aligned} \ln p(X) - KL[Q(z|X)||p(z|X)] &= \mathbb{E}_{z\sim Q}[\ln Q(z|X) - \ln p(X|z) - \ln p(z)] \\ &= \mathbb{E}_{z\sim Q}[\ln p(X|z) + \ln \frac{p(z)}{Q(z|X)}] \\ &= \mathbb{E}_{z\sim Q}[\ln p(X|z)] + \mathbb{E}_{z\sim Q}[\ln \frac{p(z)}{Q(z|X)}] \\ &= \mathbb{E}_{z\sim Q}[\ln p(X|z)] - KL[Q(z|X)||p(z)] \end{aligned} \tag{5}$$

위의 식은 VAE의 가장 중요한 부분이라 할 수 있겠다. 우리는 결국 분포 $Q(z|X)$와 $p(z|X)$를 근사해야만 하는 목적이 있다. 즉, 두 분포의 거리가 줄어야 되는 셈인데, 이는 ***식(4)***를 최소화해야 한다고 이해할 수 있겠다. 그렇다면 ***식(5)***에서 왼쪽은 $p(X)$가 이미 고정된 값이므로, 오른쪽 부분이 최대가 되어야 한다는 말과 동일하게 된다. 따라서 우리는 새로운 목적이 생겼다. 오른쪽 식을 최대값이 되게 하는 $Q(z|X)$를 찾아야 한다. 여기서 VAE는 또 다른 가정을 한다.

$$Q(z|X) = \mathcal{N}(z|\mu(X;\phi), \Sigma(X;\phi)) \tag{6}$$

μ와 Σ는 ϕ를 parameter, X를 domain으로 가지는 deterministic function이라 할 수 있겠다. 2.1에서 설명한 $f(z;\theta)$를 떠올린다면 이 함수들 역시 결국 뉴럴넷으로 모델링이 될 것이라는 점, ϕ는 backpropagation을 통해 최적화가 될 것이라는 점을 예상해볼 수 있다. 단, Σ는 Diagonal Matrix로 제한을 둔다. 이런 가정을 두는 이유는 계산이 편하기 때문이다. 우리가 VAE에서 둔 두 개의 가정 $p(Z) \sim \mathcal{N}(0, I)$과 $Q(z|X) \sim \mathcal{N}(z|\mu(X;\phi), \Sigma(X;\phi))$을 이용하면, ***식(5)***에서 오른쪽 부분 중의 KL divergence는 다음과 같이 표현이 된다.

$$\begin{aligned} KL[Q(z|X)||p(z)] &= KL[\mathcal{N}(\mu(X;\phi), \Sigma(X;\phi))||\mathcal{N}(0, I)] \\ &= \frac{1}{2}\Big[Tr(\Sigma(X;\phi)) + \mu(X;\phi)^T - D - \ln|\Sigma(X;\phi)|\Big] \end{aligned} \tag{7}$$

식(5)에서 오른쪽 첫번째 부분은, 만약 우리가 모든 샘플 $z \in Z(\neq Z')$ 을 앞서 정의한 $f(z;\theta)$를 통해 계산을 할 수도 있겠지만 너무 시간이 많이 걸린다. 따라서 하나의 샘플 $z \in Z$를 뽑고, $p(X|z)$를 $\mathbb{E}_{z\sim Q}[\ln p(X|z)]$으로 가정하도록 한다 (Stochastic Gradient Descent). 따라서 ***식(5)***는 다음과 같이 수학적으로 다시 풀어쓸 수 있다.

$$\mathbb{E}_{X\sim Data}\Big[\ln p(X) - KL[Q(z|X)||p(z|X)]\Big] = \mathbb{E}_{X\sim Data}\Big[\mathbb{E}_{z\sim Q}[\ln p(X|z)] - KL[Q(z|X)||p(z)]\Big] \tag{8}$$

따라서 위 식에 gradient를 구하게 되면 $\mathbb{E}$는 선형결합이므로 gradient는 $\mathbb{E}$ 안으로 들어갈 수 있게된다. 따라서, 우리는 단일 샘플 $x \in X$와 $Q(z|X)$에서 얻을 수 있는 단일 잠재변수 z를 통해서 $\ln p(X|z) - KL[Q(z|X)||p(z)]$를 계산할 수 있다. 그 다음 모든 값들에 대해서 평균을 취하게 되면 ***식(8)***로 수렴하게 된다.

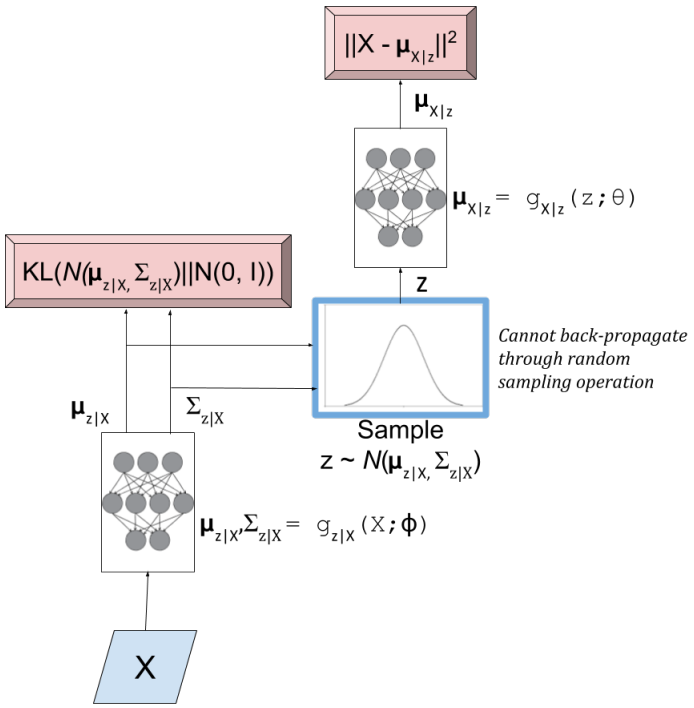

Figure 2

그러나 식을 자세히 들여다보면, $\mathbb{E}_{z\sim Q}[\ln p(X|z)]$에서 z는 $Q(z|X)$에 의해서 샘플링 되는데, 식은 이 연관성을 제대로 반영하지 못하는 것을 할 수 있다*[Figure 2]*. 다르게 바라보자면, 최적 ϕ를 구하기 위해 gradient를 도출하는 과정(backpropagation)은 deterministic function에서만 이루어지는데 샘플링이라는 것은 determinisitc이 아니다. 즉, Backpropagation 을 위해 Stochastic input에 대해서 Stochastic gradient descent을 사용한다고 해서 함수자체가 sstochastic해서는 안된다. 따라서 이 문제를 풀기 위해서는 *reparameterization trick*을 사용한다.

Reparametrization trick은 샘플링에 대한 작업을 input level로 바꿔준다. 우리가 필요한 분포는 $\mathcal{N}(\mu(X;\phi),\Sigma(X;\phi))$이지만, 이 Gaussian Dstribution은 선형결합으로 아래와 같이 표현이 될 수 있다.

$$\mathcal{N}(\mu(X;\phi),\Sigma(X;\phi)) = \mu(X;\phi) + \Sigma(X;\phi)^{\frac{1}{2}} \times \mathcal{N}(0,I) \tag{9}$$

즉, 샘플링은 ϵ $\mathcal{N}(0,I)$에서 이루어지고, encoder에서 구한 $\mu(X;\phi),\Sigma(X;\phi)$의 값은 선형결합만을 통해서 $Q(z|X)$의 분포에서 샘플링을 한 결과와 동일하게 만들어줄 수 있다. 따라서 샘플링은 input level에서 이루어지고, deterministic한 선형결합만이 모델에 남아있게 되므로 gradient descent를 충분히 수행할 수 있게 된다. 따라서 식(8)은 또 다시 다음과 같이 표현이 된다.

$$\begin{aligned}\mathbb{E}_{X\sim Data}\Big[\mathbb{E}_{z\sim Q}[\ln p(X|z)] - KL[Q(z|X)||p(z)]\Big] \\ = \mathbb{E}_{X\sim Data}\Big[\mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}[\ln p(X|z=\mu(X)+\Sigma(X;\phi)^{\frac{1}{2}}\times\epsilon)] \\ - KL[Q(z|X)||p(z)]\Big]\end{aligned} \tag{10}$$

따라서, 고정된 X와 ϵ으로 deterministic/continous한 함수를 stochastic gradient descent를 통해 back-propagation을 진행하고 이를 통해 최적 θ와 ϕ를 구할 수 있게 된다*[Figure 3]*.

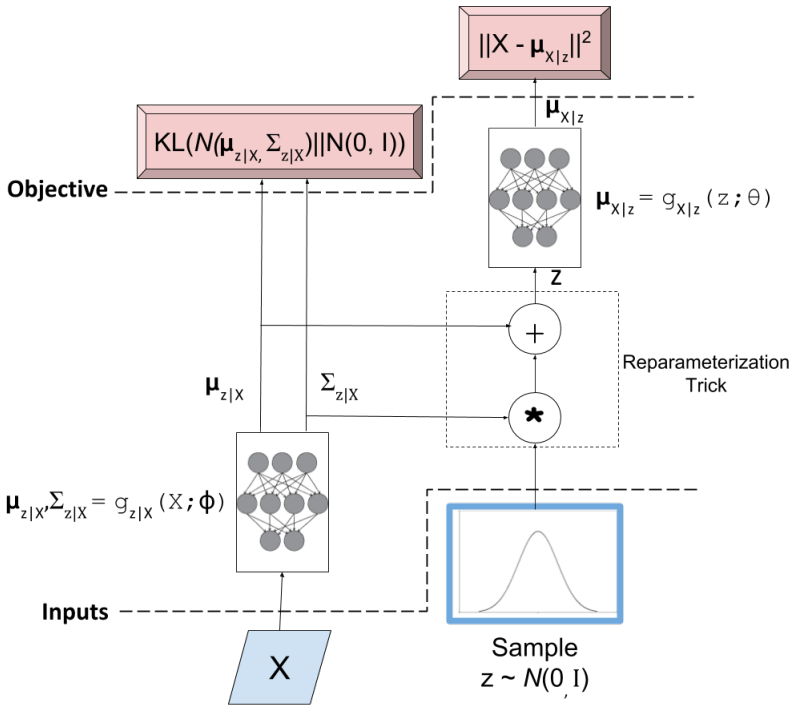

Figure 3

APPENDIX

If we consider some Bayesian model, we have seen that there exist some observed variables(or can be denoted as evidence E), latent variables(or hidden H) and model parameters. This model represents the joint distribution of evidence and hidden variable such as $p(E, H)$. The main goal of our tasks is to find out the conditional distribution over the hidden variable H given our evidence E, $p(H|E)$, and the distribution of evidence, $p(E)$. To achieve our goal, we need to compute the following equation (but known as Bayes theorem. Also note that it is enough to show the discrete case WLOG):

$$p(H|E) = \frac{p(E|H) \times p(H)}{p(E)} = \frac{p(E|H) \times p(H)}{\sum_H p(E, H)} \tag{11}$$

We only have some evidence so that we need to compute all the possible choice of hidden variable to find $p(E)$. Unfortunately if hidden variable has very high dimension, how can we finish our job in a given time. Previously, EM was suggested to infer $p(H|E)$ however if there are more cases which we cannot find the posterior distribution. Thus there are other possible methods to solve this complicated problems, MCMC(Markov Chain Monte Carlo) and VI(Variational Inference). This homework will only focus on VI, but based on what I have acknowledged about MCMC, VI can handle bigger data with faster speed. After deriving the key concept of VI, the difference between VI and EM will be reviewed quickly and variational MoG will be covered.

[Variational Inference]

Thinking about ELBO(evidence lower bound) is a good starting point of variational inference. Like how we did in EM, $\ln p(E)$ is decomposed as follows (Note that the model parameter θ does not appear because the parameters are now stochastic variables and are absorbed into H On the other hand, λ is the variational parameter which comes with the variational distribution Q. From the perspective of calculus of variations, λ will be used to optimize the functional.):

$$
\begin{aligned}
\ln p(E) &= \ln \sum_H p(H,E) = \ln \sum_H Q(H|E,\lambda)\frac{p(H,E)}{Q(H|E,\lambda)} \\
&\geq \sum_H Q(H|E,\lambda) \ln \frac{p(H,E)}{Q(H|E,\lambda)} \\
&= \sum_H Q(H|E,\lambda)\ln p(H,E) - Q(H|E,\lambda)\ln Q(H|E,\lambda) = \mathcal{L}(\lambda) \\
&= \mathbb{E}_Q[\ln p(E|H)] - KL[Q(H|E,\lambda)||p(H|E)]
\end{aligned} \tag{12}
$$

We have already known that if we set $Q(H|E,\lambda) = p(H|E)$, KL divergence is zero so that ELBO($\mathcal{L}(\lambda)$) term can be same as $\ln p(E)$. As mentioned before, however, what if we cannot find the posterior distribution $p(H|E)$. Thus we need some assumption toward the distribution Q for inferring ELBO, which is mean field theory:

$$
Q(H) = \prod_{i \leq |H|} q_i(H_i|\lambda_i) \tag{13}
$$

This means that Hidden variables with respect to Q don't have any dependency each others. It is a quite interesting fact since in our original model, there might be any dependency in terms of hidden variables (even though we will never know about). But the introduced distribution Q does not care about those dependency. This mean field variational approximation will make our life easier. Meanwhile, it should be empathized that Q is unknown so that it can be any distribution. Thus, any well known distribution can be introduced as far as we can find a good parameter λ to approximate the real distribution. This is the reason we will use conjugate prior. Then, let's look at ELBO closely keeping in mind about the mean field theorem:

$$
\begin{aligned}
\mathcal{L}(\lambda) &= \sum_H \big[Q(H|E,\lambda)\ln p(H,E) - Q(H|E,\lambda)\ln Q(H|E,\lambda)\big] \\
&= \sum_H \big[\prod_{i\leq|H|} q_i(H_i|E,\lambda_i)\ln p(H,E) - \prod_{i\leq|H|} q_i(H_i|E,\lambda_i)\ln \prod_{i\leq|H|} q_i(H_i|E,\lambda_i)\big] \\
&= \sum_H \big[\prod_{i\leq|H|} q_i(H_i|E,\lambda_i)\ln p(H,E) - \prod_{i\leq|H|} q_i(H_i|E,\lambda_i)\sum_{i\leq|H|}\ln q_i(H_i|E,\lambda_i)\big]
\end{aligned} \tag{14}
$$

Now we can optimize a single variational parameter λ_j by regarding others $\lambda_{\neg j}$ as a constant. In the end q_j will be a well approximated functional to the true distribution. Thus,

$$
\begin{aligned}
\mathcal{L}(\lambda_j) &= \sum_H \big[\prod_{i\leq|H|} q_i(H_i|E,\lambda_i)\{\ln p(H,E) - \sum_{k\leq|H|}\ln q_k(H_k|E,\lambda_k)\}\big] \\
&= \sum_{H_j}\sum_{H_{\neg j}}\big[q_j(H_j|E,\lambda_j)\prod_{i\leq|H|,i\neq j} q_i(H_i|E,\lambda_i)\{\ln p(H,E) - \sum_{k\leq|H|}\ln q_k(H_k|E,\lambda_k)\}\big] \\
&= \sum_{H_j}\sum_{H_{\neg j}} q_j(H_j|E,\lambda_j)\prod_{i\leq|H|,i\neq j} q_i(H_i|E,\lambda_i)\ln p(H,E) \\
&\quad - \sum_{H_j}\sum_{H_{\neg j}} q_j(H_j|E,\lambda_j)\prod_{i\leq|H|,i\neq j} q_i(H_i|E,\lambda_i)\sum_{k\leq|H|}\ln q_k(H_k|E,\lambda_k) \\
&= \sum_{H_j}\sum_{H_{\neg j}} q_j(H_j|E,\lambda_j)\prod_{i\leq|H|,i\neq j} q_i(H_i|E,\lambda_i)\ln p(H,E) \\
&\quad - \sum_{H_j}\sum_{H_{\neg j}} q_j(H_j|E,\lambda_j)\prod_{i\leq|H|,i\neq j} q_i(H_i|E,\lambda_i)\{\sum_{k\leq|H|,k\neq j}\ln q_k(H_k|E,\lambda_k) + \ln q_j(H_j|E,\lambda_j)\} \\
&= \sum_{H_j} q_j(H_j|E,\lambda_j)\big[\sum_{H_{\neg j}}\prod_{i\leq|H|,i\neq j} q_i(H_i|E,\lambda_i)\ln p(H,E)\big] \\
&\quad - \sum_{H_j} q_j(H_j|E,\lambda_j)\big[\sum_{H_{\neg j}}\prod_{i\leq|H|,i\neq j} q_i(H_i|E,\lambda_i)\{\sum_{k\leq|H|,k\neq j}\ln q_k(H_k|E,\lambda_k) + \ln q_j(H_j|E,\lambda_j)\}\big]
\end{aligned} \tag{15}
$$

We only concern about λ_j , however, $\sum_{H_{\neg j}} \prod_{i \leq |H|, i \neq j} q_i(H_i|E, \lambda_i) \sum_{k \leq |H|, k \neq j} \ln q_k(H_k|E, \lambda_k)$does not contain anything about λ_j. Therefore $\sum_{H_j} \sum_{H_{\neg j}} \prod_{i \leq |H|, i \neq j} q_i(H_i|E, \lambda_i) \sum_{k \leq |H|, k \neq j} \ln q_k(H_k|E, \lambda_k)$ has same value but it can be regarded as constant C. If we define a new distribution

$$\log \tilde{p}(H, E) = \sum_{H_{\neg j}} \prod_{i \leq |H|, i \neq j} q_i(H_i|E, \lambda_i) \ln p(H, E) \tag{16}$$

, ELBO should be as follows:

$$\begin{aligned}
\mathcal{L}(\lambda_j) &= \sum_{H_j} q_j(H_j|E, \lambda_j) \big[\sum_{H_{\neg j}} \prod_{i \leq |H|, i \neq j} q_i(H_i|E, \lambda_i) \ln p(H, E) \big] - \sum_{H_j} q_j(H_j|E, \lambda_j) \ln q_j(H_j|E, \lambda_j) + C \\
&= \sum_{H_j} q_j(H_j|E, \lambda_j) \ln \tilde{p}(H, E) - \sum_{H_j} q_j(H_j|E, \lambda_j) \ln q_j(H_j|E, \lambda_j) + C \\
&= \sum_{H_j} q_j(H_j|E, \lambda_j) \ln \frac{\tilde{p}(H, E)}{q_j(H_j|E, \lambda_j)} + C \\
&= C - KL[q_j(H_j|E, \lambda_j) || \tilde{p}(H, E)]
\end{aligned} \tag{17}$$

Therefore, we can maximize ELBO by let $q_j(H_j|E, \lambda_j)$ be same with $\tilde{p}(H, E)$. However when it comes to $\tilde{p}(H, E)$, it can be show that

$$\begin{aligned}
\tilde{p}(H, E) &= \sum_{H_{\neg j}} \prod_{i \leq |H|, i \neq j} q_i(H_i|E, \lambda_i) \ln p(H, E) \\
&= \mathbb{E}_{q_{i \neq j}}[\ln p(H, E)] + C
\end{aligned} \tag{18}$$

which means an expectation with respect to the q distributions over all variables H_i for $i \neq j$. Therefore we obtain a general expression for the optimal solution:

$$\ln q_j^*(H_j|E, \lambda_i) = \mathbb{E}_{q_{i \neq j}}[\ln p(H, E)] + C \tag{19}$$

References

[1] Carl Doersch. *Tutorial on Variational Autoencoders.*

[2] http://bjlkeng.github.io/posts/variational-autoencoders/

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning.*

[4] Seungjin Choi. *Lecture Note.*