1 Entropy and KL Divergence

1.1 Entropy

유한집합에서 값을 갖는 확률변수 X의 Entropy는 X의 각각의 값 x의 Shannon information의 평균값으로 정의한다.

$$H(X) = \sum_x P(x) \log \frac{1}{P(x)}$$

여기서 만약 $P(x) = 0$ 이면 $H(x) = 0$ 이라고 계산한다. 이 때 $H(x) \geq 0$ 이며, 등호는 X 가 상수이며 그에 대한 확률이 1 일 때 성립한다. 또 다른 information을 다음과 같이 정의한다. (보통 log 로서 자연로그를 사용한다. 여기에서는 $\log_2$ 를 사용한다고 가정하고 예시를 계산한다.)

$$h(x) = P(x) \log \frac{1}{P(x)}$$

Joint Entropy는 다음과 같이 정의한다.

$$H(X, Y) = \sum_{x,y} P(x, y) \log \frac{1}{P(x, y)}$$

만약 두 확률변수 X, Y가 독립이라면 $P(x, y) = P(x)P(y)$에서 다음을 얻을 수 있다.

$$H(x, y) = H(x) + H(y)$$

Entropy는 확률변수의 평균 정보량이며, 확률변수의 불확실성을 나타낸다고 한다. 따라서 위에서 언급했다시피 $P(X = x) = 1$일 때는 불확실성이 0이므로 $H(X) = 0$ 이다.

예를 들어 집합 {0, 1, 2} 에서 값을 갖는 확률변수 X는 동전 하나를 던져서 앞면이 나오면 $X = 0$이고, 뒷면이 나오면 동전을 다시 던져 앞면이 나오면 $X = 1$, 뒷면이 나오면 $X = 2$ 이다. 동전이 fair coin이라 할 때 다음과 같다.

$$P(0) = \frac{1}{2}$$
$$P(1) = P(2) = \frac{1}{4}$$

따라서 다음과 같다.

$$H(X) = \frac{1}{2} \log 2 + \frac{1}{4} \log 4 + \frac{1}{4} \log 4 = \frac{3}{2}$$

위의 전체 과정을 다음과 같이 나눠서 생각할 수 있다.

$$H\left(\frac{1}{2}, \frac{1}{2}\right) = \frac{1}{2} \log 2 + \frac{1}{2} \log 2 = 1$$

다음 과정은 위의 entropy의 절반이다. 따라서 전체 entropy는 이들의 합이다.

$$H(X) = H\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{1}{2} H\left(\frac{1}{2}, \frac{1}{2}\right) = \frac{3}{2}$$

1.2 Cross Entropy

두 확률분포 $P(x)$와 $Q(x)$가 주어졌다고 하자. 실제 분포는 P를 따르는 것을 추정한 것이 Q라면 평균을 낼 때 $Q(x)$ 대신 확률 $P(x)$를 사용해야 한다. 즉, 다음과 같다.

$$H_P(Q) = H(P,Q) = -\sum_x P(x)\log Q(x)$$

이를 실제 분포 (true distribution) P에 대한 Q의 Cross Entropy라고 한다. ML에서는 실제 분포를 구하려고 NN을 사용해서 만든 것의 분포가 Q일 때 cross entropy를 minimize해서 P에 가깝도록 한다. 즉, 데이터는 true distribution이라고 하고 learning에서 얻은 분포를 Q라고 하면, 이들의 Cross Entropy를 minimize 한다.

Classification에 대한 예시 하나를 고려해 보자. Data $\{x_i\}$에 대해서 집합 $0, 1$ 안에서 값을 갖는 target y_i 를 생각하자. training 데이터에서의 y_i 는 확률분포 $P(y_i|x_i)$ 이며 학습을 통해 얻은 분포는 $Q(y_i|x_i)$ 라고 할 수 있다. 이 때 Loss 함수로서 cross entropy를 사용한다.

$$L = -\sum_{i=1}^{N}\sum_{y_i} P(y_i|x_i)\log Q(y_i|x_i) = -\sum_{i=1}^{N}[P(1|x_i)\log Q(1|x_i) + P(0|x_i)\log Q(0|x_i)]$$

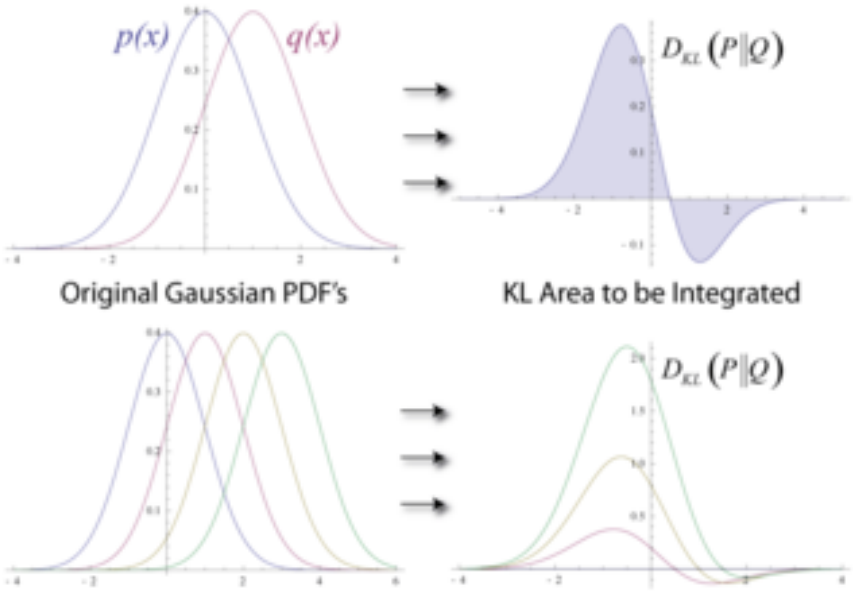
즉, Binary Classification의 경우에 Cross entropy 최소화는 MLE를 계산하는 것 (Likelihood 최대화)와 같은 의미이다:

$$\begin{aligned} f_p(Y=y) &= p^y(1-p)^{1-y} \\ LE &= \Pi_i^N f_p(Y=y_i) = \Pi_i^N p^{y_i}(1-p)^{1-y_i} \\ \therefore \quad -\log LE &= \text{Entropy} \end{aligned}$$

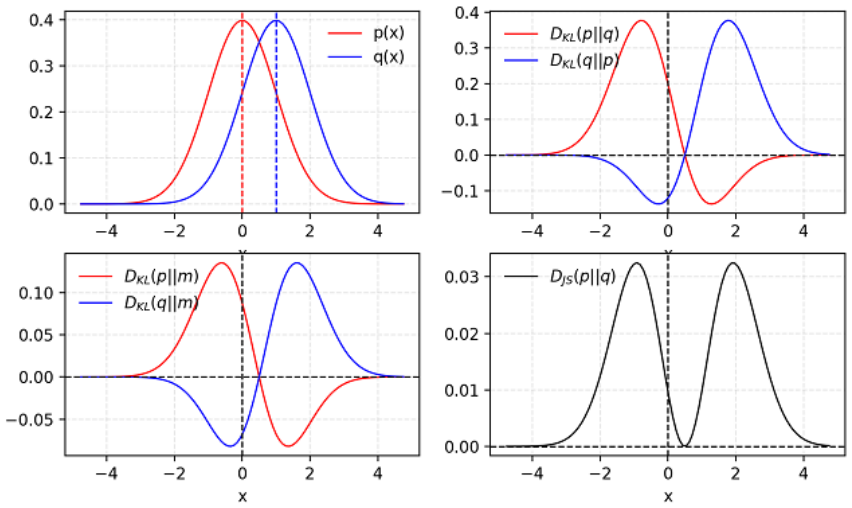
1.3 KL Divergence

두 가지 확률분포 P와 Q를 비교하는 방법으로 많이 쓰이는 것에는 Kullback-Leibler divergence와 Jensen-Shannon divergence가 있다. KL Divergence는 다음과 같다.

$$\begin{aligned} D_{KL}(P||Q) &= E_{x\sim P}\left[\log\frac{P(x)}{Q(x)}\right] \\ &= -\sum_i P(x_i)\log\frac{Q(x_i)}{P(x_i)} \\ &= \sum_i P(x_i)\log P(x_i) - \sum_i P(x_i)\log Q(x_i) \\ &= -H(P) + H(P,Q) \geq 0 \end{aligned}$$



위의 그림에서 볼 수 있다시피, 두 확률변수가 '멀어질수록' KL Divergence가 커진다. 즉, KL Divergence는 확률변수간 '어떤 간격'을 나타낸다. 다만, 이는 metric도 아니고, symmetric하지도 않다.



KL Divergence에는 치명적인 약점이 있는데, 고차원 분포에서, 겹치는 부분이 없는 저차원 분포 사이의 divergence는 0이 된다. 이렇게 되면 Loss function이 계속 무한대라서 학습이 불가하다.

VAE 에서의 변분추론(Variational Inference, VI)에 사용하기 위해 KL Divergence 의 식을 변형하면 다음과 같다. true posterior $p(z|x)$ 에 근사한 $q(z)$ 를 만들기 위해 KL Divergence 를 줄이는 방식으로 q 의 parameter 를 업데이트

한다는 것이 VI 의 아이디어이다:

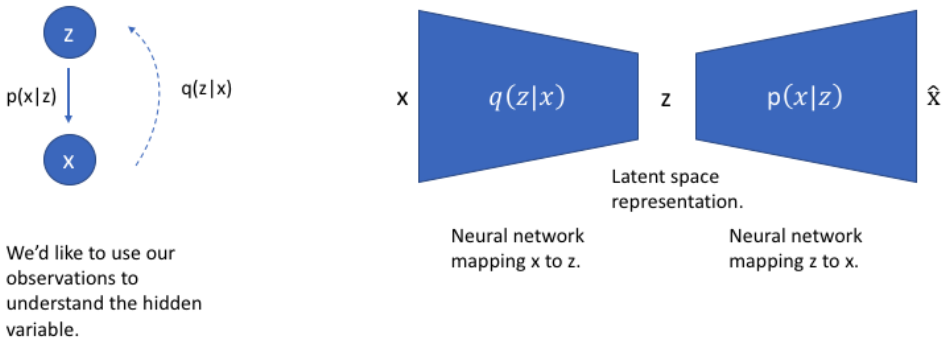
$$\begin{aligned} D_{KL}\big(q(z)||p(z|x)\big) &= \int q(z)\log\frac{q(z)}{p(z|x)}dz \\ &= \int q(z)\log\frac{q(z)p(x)}{p(x|z)p(z)}dz \\ &= \int q(z)\log\frac{q(z)}{p(z)}dz + \int q(z)\log p(x)dz - \int q(z)\log p(z|x)dz \\ &= D_{KL}\big(q(z)||p(z)\big) + \log p(x) - \mathbb{E}_{z\sim q(z)}\big[\log p(x|z)\big] \end{aligned} \tag{1}$$

summary Classifier를 만들기 위한 많은 기계학습(autoencoder, GAN 등)에서는 training data가 따르는 분포가 하나 정해져 있다고 가정한다. 이것이 $P(x)$이다. 한편 우리가 학습시키는 기계는 $Q(x)$를 답으로 준다. 따라서 기계학습의 목표는 $Q(x)$를 움직여서 $P(x)$에 최대한 가깝게 되도록 하는 것이고 이를 위해서 Loss 함수를 정의하는데, 여기서 Entropy를 사용하거나 KL Divergence를 사용한다.

2 Variational Auto Encoder (VAE)

2.1 Model Form

논문의 내용을 하나하나 뜯어보기 전에 VAE 의 전체적인 구조를 대략적으로 알아보자. VAE 는 AutoEncoder(AE)를 응용한 하나의 예시로, latent space 를 AE 와는 약간 다르게 이용한다. AE 의 latent space 에서 추출되는 벡터는 각각이 어떤 압축된 정보를 담고 있었다. 하지만 VAE 는 확률분포에 근거해서 새로운 값들(AE 의 latent vector 에 해당하는 것)을 latent space 에서 만들어줘야 하기 때문에 '평균들' 과 '분산들' 을 뽑아내준다.



latent vector 를 z 라고 두고, 데이터를 x 라고 둘 것이다. ($\hat{x}$ 는 x 를 다시 생성하는 것이기에 구분하지 않는다.) ϕ 를 model parameter 라고 했을 때 encoder 는 '데이터 x 가 주어졌을 때 latent vector 를 생성하는 것이며 $q_\phi(z|x)$ (true posterior $p_\theta(z|x)$ 의 근사) 라는 확률과정으로 표현할 수 있다. decoder $p_\theta(x|z)$ 는 latent vector z 가 주어졌을 때 원본이미지를 다시 생성하는 확률과정이다.

2.2 Loss

논문에서는 marginal likelihood (bayesian 에서 evidence 를 종종 marginal likelihood 라고 하기도 한다.) $p_\theta(x)$ 를 최대화하도록 하면 이런저런 유도과정을 거쳐 1. $q_\phi(z|x)$ 가 $p_\theta(z|x)$ 에 가까워지며 2. z 로부터 원본데이터 x 를 얻는 확률도 높아지게 된다고 설명하고 있다.

각각의 데이터 index 를 $i = 1, 2, ..., N$ 라고 두면 marginal likelihood 는 다음과 같다.

$$\log p_\theta(x^1,\, x^2,\, \cdots,\, x^N) = \sum_{i=1}^{N} \log p_\theta(x^i)$$

또한 각각의 데이터에 대한 marginal likelihood 는 식 (1) 을 이용하면 다음과 같이 전개될 수 있다.

$$\begin{aligned}\log p_\theta(x^i) &= D_{KL}\big(q_\phi(z)||p_\theta(z|x^i)\big) - D_{KL}\big(q_\phi(z)||p_\theta(z)\big) + \mathbb{E}_{z\sim q_\phi(z)}\big[\log p_\theta(x^i|z)\big] \\ &= D_{KL}\big(q_\phi(z|x^i)||p_\theta(z|x^i)\big) - D_{KL}\big(q_\phi(z|x^i)||p_\theta(z)\big) + \mathbb{E}_{z\sim q_\phi(z|x^i)}\big[\log p_\theta(x^i|z)\big]\end{aligned}$$

이 때 첫번째 항은 항상 0 이상(∵ KL Divergence)이기 때문에 다음과 같이 나타낼 수 있다:

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

이 때 $\mathcal{L}(\theta, \phi; x^i)$ 를 ELBO (Evidence Lower BOund) 라고 부르기도 한다. 여기서 잠시 이 식을 해석해보자면, maximum likelihood $\log p_\theta(x^i|z)$ 문제를 푸는데, 거기에 variational approximation $q_\phi(z|x^i)$ 와 $p_\theta(z)$ 가 가까워지도록 하는 regularization 항을 추가한 것으로 볼 수 있다. 우리의 목적은 ELBO 를 최대화 하는 것이다. 따라서 최소화해야하는 손실함수(Loss)는 다음과 같다:

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

true prior $p_\theta(z) = N(0, \mathbb{I})$ 이고 posterior approximation $q_\phi(z|x^i)$ 가 정규분포라고 가정하면 Loss 의 첫번째 항은 다음과 같이 변환이 가능하다. J 는 z 의 차원수이다.

$$\begin{aligned}D_{KL}\big(q_\phi(z|x^i)||p_\theta(z)\big) &= \int q_\phi(z|x^i)\big(\log q_\phi(z|x^i) - \log p_\theta(z)\big)dz \\ &= \int N(z;\, \mu, \sigma^2)\log N(z;\, \mu, \sigma^2)dz - \int N(z;\, \mu, \sigma^2)\log N(z;\, 0, \mathbb{I})dz \\ &= -\left(\frac{J}{2}\log(2\pi) + \frac{1}{2}\sum_{j=1}^{J}(1 + \log\sigma_j^2)\right) + \left(\frac{J}{2}\log(2\pi) + \frac{1}{2}\sum_{j=1}^{J}(\mu_j^2 + \sigma_j^2)\right) \\ &= -\frac{1}{2}\sum_{j=1}^{J}\left(1 + \log\sigma_j^2 - \mu_j^2 - \sigma_j^2\right)\end{aligned}$$

Loss 의 두번째 항은 reconstruction loss 에 해당한다. encoder가 데이터 x 를 받아서 q 로부터 z 를 뽑는 것과, decoder 가 z 를 받아 원래 데이터 x 를 복원하는 것의 (negative) cross-entropy 를 의미한다.

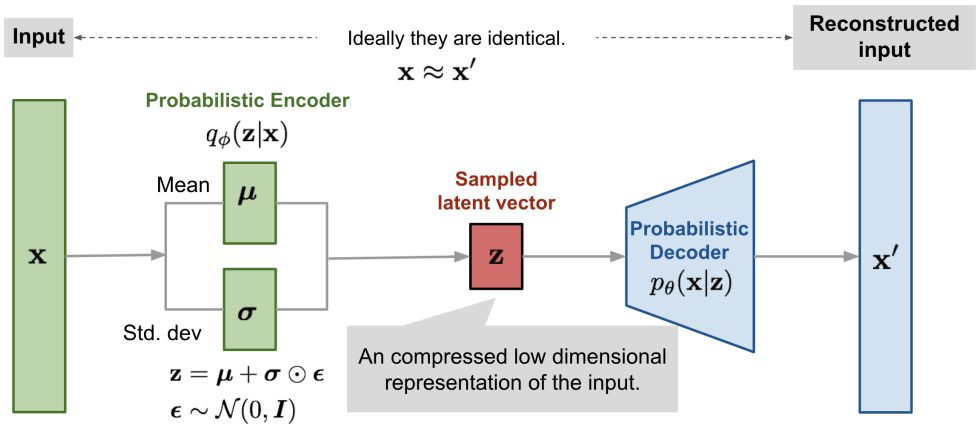
$z^(i,l) \sim q_\phi(z|x^i)$ 은 미분가능성을 위해 다음과 같이 sampling 된다:

$$\begin{aligned}z^{(i,l)} &= g_\phi(x^i,\, \epsilon^l) = \mu^i + \sigma^i\epsilon^l \\ \text{where } \epsilon^l &\sim N(0, \mathbb{I})\end{aligned}$$

따라서 Monte Carlo estimate 에 의하면 Loss 의 두번째 항은 다음과 같이 계산될 수도 있다:

$$\mathbb{E}_{z\sim q_\phi(z|x^i)}\big[\log p_\theta(x^i|z)\big] = \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(x^i|z^{(i,l)})$$

코드에서는 KL Divergence 를 전개한 것과 negative cross entropy 를 사용한다. 위의 과정을 요약하면 다음과 같이 사진으로 표현할 수 있겠다.


Input
Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Reconstructed input
Probabilistic Encoder
$q_\phi(\mathbf{z}|\mathbf{x})$
Mean
$\boldsymbol{\mu}$
$\boldsymbol{\sigma}$
Std. dev
$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$
$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$
$\mathbf{x}$
Sampled latent vector
$\mathbf{z}$
An compressed low dimensional representation of the input.
Probabilistic Decoder
$p_\theta(\mathbf{x}|\mathbf{z})$
$\mathbf{x}'$