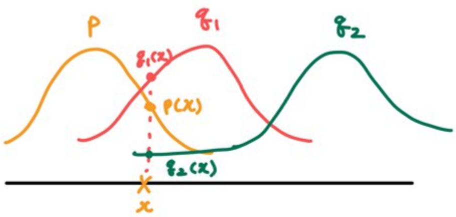
6. Kullback-Leibler Divergence

Kullback-Leibler Divergence

두 확률분포 p와 q의 다름(dissimilarity)를 측정한 것으로 KLD는 비대칭이기 때문에 거리라고 부르지 않는다.

$$\begin{aligned} KL(p||q) &= -\mathbb{E}_{x\sim p(x)}\left[\log\frac{q(x)}{p(x)}\right] \\ &= -\int p(x)\log\frac{q(x)}{p(x)}dx \end{aligned}$$

덜 비슷할수록 큰 값을 가지고, 완전 똑같게 되면 0이 된다.(세 개의 분포를 비교한 그림을 통해 알 수 있다.)



$$= -\int p(x)\log\frac{q(x)}{p(x)}dx$$

$$\Rightarrow \frac{q_1(x)}{p(x)} > \frac{q_2(x)}{p(x)}$$

$$\log\frac{q_1(x)}{p(x)} > \log\frac{q_2(x)}{p(x)}$$

$$-\log\frac{q_1}{p} < -\log\frac{q_2}{p}$$

DNN Optimization using KL-Divergence

실제 확률분포와 신경망의 확률분포를 최대한 같게 하는 것이 우리의 목적이므로, KL-Divergence를 최소화하는 파라미터를 찾는 방식으로 생각할 수 있다.

즉, 손실함수를 KL-Divergence로 잡아도 된다는 이야기로 똑같이 Gradient descent를 통해서 적절한 파라미터를 찾을 수 있다.

θ를 가지는 neural network 확률분포

$$\mathcal{L}(\theta) = -\mathbb{E}_{x\sim p(x)}\left[\mathbb{E}_{y\sim p(y|x)}\left[\log\frac{p_\theta(y|x)}{p(y|x)}\right]\right]$$

실제 확률분포

→ $KL(P(y|x) \| P_\theta(y|x))$

by Monte-Carlo (확률분포에서 샘플링을 통해 근사하는 것)

$$D = \{(x_i, y_i)\}_{i=1}^N$$

→ $x_i \sim P(x)$

$y_i \sim P(y|x = x_i)$

$$\begin{aligned}\mathcal{L}(\theta) &\approx -\frac{1}{N\cdot k}\sum_{i=1}^{N}\sum_{j=1}^{k}\log\frac{p_\theta(y_{i,j}|x_i)}{p(y_{i,j}|x_i)} \\ &\approx -\frac{1}{N}\sum_{i=1}^{N}\log\frac{p_\theta(y_i|x_i)}{p(y_i|x_i)}, \text{if } k = 1.\end{aligned}$$

K=1 이라 가정, x 하나에 대해 y 하나 대응

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{p_\theta(y_i|x_i)}{p(y_i|x_i)}$$

$$\hat{\theta} = \underset{\theta\in\Theta}{\operatorname{argmin}}\,\mathcal{L}(\theta)$$

$$\theta \leftarrow \theta - \alpha\nabla_\theta\mathcal{L}(\theta)$$