# Mathematical Foundation of DNN : HW 11

Jeong Min Lee

June 10, 2024

## 1

### a

$$\begin{aligned}
\mathrm{VLB}^{(K)}_{\theta,\phi}(x) &= \mathbb{E}_{Z_1,\cdots,Z_k\sim q_\phi(z|x)}\left[\log \frac{1}{K}\sum_{k=1}^{K}\frac{p_\theta(x|Z_k)p_Z(Z_k)}{q_\phi(Z_k|x)}\right] \\
&\le \log\left(\mathbb{E}_{Z_1,\cdots,Z_k\sim q_\phi(z|x)}\left[\frac{1}{K}\sum_{k=1}^{K}\frac{p_\theta(x|z_k)p_Z(z_k)}{q_\phi(z_k|x)}\right]\right) \\
&= \log\left(\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{Z\sim q_\phi(z|x)}\left[\frac{p_\theta(x|z)p_Z(z)}{q_\phi(z_k|x)}\right]\right) \\
&= \log\left(\frac{1}{K}\sum_{k=1}^{K}p_\theta(x)\right) \\
&= \log p_\theta(x)
\end{aligned}$$

### b

In this problem, I denoted the $M$ dimensional continuous uniform distribution as $\mathcal{U}(1,K)$

$$\begin{aligned}
\mathrm{VLB}^{(K)}_{\theta,\phi}(x) &= \mathbb{E}_{Z_1,\cdots,Z_k\sim q_\phi(z|x)}\left[\log \frac{1}{K}\sum_{k=1}^{K}\frac{p_\theta(x|Z_k)p_Z(Z_k)}{q_\phi(Z_k|x)}\right] \\
&= \mathbb{E}_{Z_1,\cdots,Z_k\sim q_\phi(z|x)}\left[\log\left(\mathbb{E}_{\{i_1,\cdots,i_M\}\sim\mathcal{U}(1,K)}\left[\frac{1}{M}\sum_{j=1}^{M}\frac{p_\theta(x|Z_{i_j})p_Z(Z_{i_j})}{q_\phi(Z_{i_j}|x)}\right]\right)\right] \\
&\ge \mathbb{E}_{Z_1,\cdots,Z_k\sim q_\phi(z|x)}\left[\mathbb{E}_{\{i_1,\cdots,i_M\}\sim\mathcal{U}(1,K)}\left[\log\left(\frac{1}{M}\sum_{j=1}^{M}\frac{p_\theta(x|Z_{i_j})p_Z(Z_{i_j})}{q_\phi(Z_{i_j}|x)}\right)\right]\right] \\
&= \mathbb{E}_{Z_{i_1},\cdots,Z_{i_M}\sim q_\phi(z|x)}\left[\log\left(\frac{1}{M}\sum_{j=1}^{i_M}\frac{p_\theta(x|Z_j)p_Z(Z_j)}{q_\phi(Z_j|x)}\right)\right] \\
&= \mathrm{VLB}^{(M)}_{\theta,\phi}(x)
\end{aligned}$$

### c

$$\begin{aligned}
&D_{KL}\left[q_\phi(\cdot|X_i)|p_\theta(\cdot|X_i)\right] \\
&= \mathbb{E}_{Z\sim q_\phi(z|X_i)}\left[\log q_\phi(Z|X_i) - \log p_\theta(Z|X_i)\right] \\
&= \mathbb{E}_{Z_1,\cdots,Z_K\sim q_\phi(Z|X_i)}\left[\frac{1}{K}\sum_{k=1}^{K}\log q_\phi(Z_k|X_i) - \log p_\theta(Z_k|X_i)\right] \\
&= \mathbb{E}_{Z_1,\cdots,Z_K\sim q_\phi(Z|X_i)}\left[\frac{1}{K}\sum_{k=1}^{K}(\log q_\phi(Z_k|X_i) - \log p_\theta(X_i|Z_k) - \log p_Z(Z_k)) + \log p_\theta(X_i)\right] \\
&= -\mathbb{E}_{Z_1,\cdots,Z_K\sim q_\phi(Z|X_i)}\left[\frac{1}{K}\sum_{k=1}^{K}\log\frac{p_\theta(X_i|Z_k)p_Z(Z_k)}{q_\phi(Z_k|X_i)}\right] + \log p_\theta(X_i)
\end{aligned}$$

Note that the second equality hold since the the expectation value of sample mean is population average and $Z_1,\cdots,Z_K$ are $i.i.d$ sample from $q_\phi(Z|X_i)$.

According to the logic above, the following equation is hold.

$$\log p_\theta(X_i) = \mathrm{VLB}^{(K)}_{\theta,\phi}(X_i) + D_{KL}\left[q_\phi(\cdot|X_i)|p_\theta(\cdot|X_i)\right] \tag{1}$$

This implies that (1) maximizing the log likelihood is equivalent to maximizing $\mathrm{VLB}^{(K)}_{\theta,\phi}$ if the $q_\phi$ is powerful enough and (2) the meaning of powerful $q_\phi$ is the one that makes the KL-divergence zero with respect to $p_\theta$

# 2

## a

$$\begin{aligned}
\mathrm{VLB}_{\theta,\phi,\lambda}(X_i) &= \mathbb{E}_{Z\sim q_\phi(z|X_i)}\left[\log\left(\frac{p_\theta(X_i|Z)r_\lambda(Z)}{q_\phi(Z|X_i)}\right)\right] \\
&\le \log\left(\mathbb{E}_{Z\sim q_\phi(z|X_i)}\left[\frac{p_\theta(X_i|Z)r_\lambda(Z)}{q_\phi(Z|X_i)}\right]\right) \\
&= \log\int_z dz\ p_\theta(X_i|z)r_\lambda(z) \\
&\approx \log p_\theta(X_i)
\end{aligned}$$

## b

When calculating the gradient with respect to $\theta, \lambda$, it is fine not to consider the expectation. Thus, according to Monte Carlo method, the gradient with respect to $\theta, \lambda$ can be evaluated after the sampling $Z_{i,k}$ from $q_\phi(z|X_i)$.

$$\begin{aligned}
\nabla_\theta \mathrm{VLB}_{\theta,\phi,\lambda}(X_i) &= \nabla_\theta \frac{1}{K}\sum_{k=1}^{K}\log\frac{p_\theta(X_i|Z_{i,k})r_\lambda(Z_{i,k})}{q_\phi(Z_{i,k}|X_i)} \\
\nabla_\lambda \mathrm{VLB}_{\theta,\phi,\lambda}(X_i) &= \nabla_\lambda \frac{1}{K}\sum_{k=1}^{K}\log\frac{p_\theta(X_i|Z_{i,k})r_\lambda(Z_{i,k})}{q_\phi(Z_{i,k}|X_i)}
\end{aligned}$$

However, when dealing with the gradient of $\phi$, the expectation should be considered. The gradient can be evaluated by using log-derivative trick as follow. Note that this is identical to the stochastic gradient of VAE.

$$\begin{aligned}
\nabla_\phi \mathrm{VLB}_{\theta,\phi,\lambda}(X_i) &= \mathbb{E}_{Z\sim q_\phi(Z|X_i)}\left[(\nabla_\phi \log q_\phi(Z|X_i))\log\frac{p_\theta(X_i|Z)r_\lambda(Z)}{q_\phi(Z|X_i)}\right] \\
&= \frac{1}{K}\sum_{k=1}^{K}(\nabla_\phi \log q_\phi(Z|X_i))\log\frac{p_\theta(X_i|Z)r_\lambda(Z)}{q_\phi(Z|X_i)}
\end{aligned}$$

## c

Rather than using log-derivative trick, reparameterization trick would be good strategy for calculating the stochastic gradients. To simplify the notation, define $\psi_{\theta,\phi,\lambda}(z) = \log\frac{p_\theta(X_i|z)r_\lambda(z)}{q_\phi(z|X_i)}$. For $Y \sim \mathcal{N}(0, I)$,

$$Z = \mu_\phi(X_i) + \Sigma_\phi^{1/2}(X_i)Y \tag{2}$$

where $\Sigma_\phi^{1/2}$ is the diagonal matrix whose diagonal entries are square root of $\Sigma_\phi$'s diagonal. Then, the $\mathrm{VLB}_{\theta,\phi,\lambda}(X_i)$ can be rewritten as follow.

$$\mathrm{VLB}_{\theta,\phi,\lambda}(X_i) = \mathbb{E}_{Y\sim\mathcal{N}(0,I)}\left[\psi_{\theta,\phi,\lambda}(\mu_\phi(X_i) + \Sigma_\phi^{1/2}(X_i)Y)\right] \tag{3}$$

As the argument of $\psi_{\theta,\phi,\lambda}$ only depends on $\phi$, the derivative of $\psi_{\theta,\phi,\lambda}$ on $\theta, \lambda$ is simple as follow.

$$\begin{aligned}
\nabla_\theta \mathbb{E}_{Y\sim\mathcal{N}(0,I)}\left[\psi(\mu_\phi(X_i) + \Sigma_\phi^{1/2}(X_i)Y)\right] &= \mathbb{E}_{Y\sim\mathcal{N}(0,I)}\left[\nabla_\theta \log p_\theta\left(X_i\middle|\mu_\phi(X_i) + \Sigma_\phi^{1/2}(X_i)Y\right)\right] \\
\nabla_\lambda \mathbb{E}_{Y\sim\mathcal{N}(0,I)}\left[\psi(\mu_\phi(X_i) + \Sigma_\phi^{1/2}(X_i)Y)\right] &= \mathbb{E}_{Y\sim\mathcal{N}(0,I)}\left[\nabla_\lambda \log r_\lambda\left(\mu_\phi(X_i) + \Sigma_\phi^{1/2}(X_i)Y\right)\right]
\end{aligned}$$

However, for derivating with respect to $\phi$, it is more complicated than the previous case.

$$\nabla_\phi \mathbb{E}_{Y\sim\mathcal{N}(0,I)}\left[\psi(\mu_\phi(X_i) + \Sigma_\phi^{1/2}(X_i)Y)\right] = \mathbb{E}_{Y\sim\mathcal{N}(0,I)}\left[-\nabla_\phi \log q_\phi\left(\mu_\phi(X_i) + \Sigma_\phi^{1/2}(X_i)Y\right)\cdot\nabla_\phi\left(\mu_\phi(X_i) + \Sigma_\phi(X_i)^{1/2}Y\right)\right]$$

If the given distributions are inserted to the equations above, the gradients can be written as follow.

$$\nabla_\theta \mathrm{VLB}_{\theta,\phi,\lambda}(X_i) = \mathbb{E}_{Y\sim\mathcal{N}(0,I)}\left[\frac{1}{\sigma^2}(X_i - f_\theta(\mu_\phi(X_i)+\Sigma_\phi^{1/2}(X_i)Y))^T\nabla_\theta f_\theta(\mu_\phi(X_i)+\Sigma_\phi^{1/2}(X_i)Y)\right]$$

$$\nabla_\lambda \mathrm{VLB}_{\theta,\phi,\lambda}(X_i) = \mathbb{E}_{Y\sim\mathcal{N}(0,I)}\left[\begin{pmatrix} (\mu_\phi(X_i)+\Sigma_\phi^{1/2}(X_i)Y-\lambda_1)^T(\mathrm{diag}(\lambda_2))^{-1} \\ -\frac{1}{2}\lambda_2^{-1}+\frac{1}{2}(\mu_\phi(X_i)+\Sigma_\phi^{1/2}(X_i)Y-\lambda_1)^T\mathrm{diag}(\lambda_2^2)(\mu_\phi(X_i)+\Sigma_\phi^{1/2}(X_i)Y-\lambda_1)\end{pmatrix}\right]$$

$$\nabla_\phi \mathrm{VLB}_{\theta,\phi,\lambda}(X_i)$$
$$= \mathbb{E}_{Y\sim\mathcal{N}(0,I)}\left[\left(-\frac{1}{2}\Sigma_\phi^{-1}\nabla_\phi\sigma_\phi+\frac{1}{2}\sigma_\phi^{-2}\|Z_i-\mu_\phi\|^2\nabla_\phi\sigma_\phi-((Z-\mu_\phi)\cdot\sigma_\phi)\sigma_\phi^{-2}\nabla_\phi(Z-\mu_\phi)\right)\nabla_\phi\left(\mu_\phi(X_i)+\Sigma_\phi(X_i)^{1/2}Y\right)\right]$$

where $\sigma_\phi$ is the vector of $\Sigma_\phi$'s diagonals and for vector $v$, $v^k$ denotes the vector whose components are powered by $k \in \mathbb{Z}$.

## 3

By running the code below, the estimated threshold was -1174.791748046875. For that threshold, the type I error rate was approximately 1.1%, while type II error rate was 0.24%.

```
'''
Step 4: Calculate standard deviation by using validation set
'''
validation_loader = torch.utils.data.DataLoader(
    dataset=validation_dataset, batch_size=batch_size)


log_probs = []
for images, _ in validation_loader:
  images = images.view(-1, 784)
  for image in images:
    image = image.view(1,-1)
    log_probs.append(nice(image).item())


mean, std = torch.mean(torch.FloatTensor(log_probs)).item(), torch.std(torch.FloatTensor(
                                                log_probs)).item()

threshold = mean - 3*std

'''
Step 5: Anomaly detection (mnist)
'''
test_loader = torch.utils.data.DataLoader(
    dataset=test_dataset, batch_size=batch_size)

count = 0
for images, _ in test_loader:
  images = images.view(images.size(0), -1)
  for image in images:
    image = image.view(1,-1)
    if nice(image).item() < threshold:
      count +=1

print(f'{count} type I errors among {len(test_dataset)} data')

'''
Step 6: Anomaly detection (kmnist)
'''
anomaly_loader = torch.utils.data.DataLoader(
    dataset=anomaly_dataset, batch_size=batch_size)

count = 0
for images, _ in anomaly_loader:
  images = images.view(images.size(0), -1)
  for image in images:
    image = image.view(1,-1)
    if nice(image).item() > threshold:
      count +=1

print(f'{count} type II errors among {len(anomaly_dataset)} data')
```

# 4

## a

$$
\begin{aligned}
\mathcal{L}(p_A, p_B) &= \mathbb{E}_{p_A, p_B}[\text{points for B}] \\
&= \mathbb{E}[\text{points for B}|\text{A plays rock}]P[\text{A plays rock}] \\
&+ \mathbb{E}[\text{points for B}|\text{A plays scissors}]P[\text{A plays scissors}] \\
&+ \mathbb{E}[\text{points for B}|\text{A plays paper}]P[\text{A plays paper}] \\
&= \left[0 \times p_B^{(1)} + 1 \times p_B^{(2)} + (-1) \times p_B^{(3)}\right] \times p_A^{(1)} \\
&+ \left[(-1) \times p_B^{(1)} + 0 \times p_B^{(2)} + 1 \times p_B^{(3)}\right] \times p_A^{(2)} \\
&+ \left[1 \times p_B^{(1)} + (-1) \times p_B^{(2)} + 0 \times p_B^{(3)}\right] \times p_A^{(3)} \\
&= p_A^{(1)}(p_B^{(2)} - p_B^{(3)}) + p_A^{(2)}(p_B^{(3)} - p_B^{(1)}) + p_A^{(3)}(p_B^{(1)} - p_B^{(2)})
\end{aligned}
$$

Let $p_A^\star = (1/3, 1/3, 1/3)^T$, $p_B^\star = (1/3, 1/3, 1/3)^T$. Then, $\mathcal{L}(p_A^\star, p_B^\star) = 0$ Define $p_B = (x, y, 1-x-y)$ where $x + y \geq 1, x \leq 0, y \leq 0$. For that $p_B$, $\mathcal{L}(p_A^\star, p_B) = \frac{1}{3}(x + 2y - 1 + 1 - 2x - y + x - y) = 0 \leq \mathcal{L}(p_A^\star, p_B^\star)$. Furthermore, $\mathcal{L}(p_A, p_B^\star) = 0$, trivially. Thus, $\mathcal{L}(p_A^\star, p_B) \leq \mathcal{L}(p_A^\star, p_B^\star) \leq \mathcal{L}(p_A, p_B^\star)$. To show that this solutions are unique, introduce some deviation in $p_B = (1/3 + l_1, 1/3 + l_2, 1/3 + l_3)$, where $l_1 + l_2 + l_3 = 0$. Suppose that $p_A^\star$ is not the unique. Then, there exists $p_A \neq p_A^\star$ that makes $\mathcal{L}(p_A, p_B) \leq 0 = \mathcal{L}(p_A, p_B^\star)$ for every $p_B$. This implies $p_A^{(1)}(l_2 - l_3) + p_A^{(2)}(l_3 - l_1) + p_A^{(3)}(l_1 - l_2) \leq 0$ for all $l_1, l_2, l_3$ with $l_1 + l_2 + l_3 = 0$. However, if we choose $(l_1, l_2, l_3) = (1/6, 1/6, -1/3), (1/6, -1/3, 1/6), (-1/3, 1/6, 1/6)$, and succesively inserting them to the inquality above, one can get $p_A^{(1)} \leq p_A^{(2)} \leq p_A^{(3)} \leq p_A^{(1)}$, which implies $p_A^{(1)} = p_A^{(2)} = p_A^{(3)} = 1/3$. This contradicts to $p_A \neq p_A^\star$. Thus, the optimal solution $p_A^\star$ is unique. This discussion is applicable to $p_B^\star$.

## b

It is obvious that only $p_A^\star$ is obvious for A. If thinking about it qualitively, if A's strategy is biased, B can employ such bias to get more score. For quantative analysis, $\mathbb{E}_{p_A, p_B}[\text{points for B}] = -\mathbb{E}_{p_A, p_B}[\text{points for A}] = 0$, which means not only expectation score for B, that of A should be zero when B chooses optimal policy. One can check that this happens when $p_A = p_A^\star$, breifly rearranging $\mathcal{L}(p_A, p_B)$.