Advanced Machine Learning – Deep Generative Models Exercise Sheet 03

VAE & GAN

Problem 1: Below we show pseudocode for implementing 3 autoencoder-like neural net architectures. The observed data is denoted as $\boldsymbol{x} \in \mathbb{R}^D$. Here, $g_{\boldsymbol{\lambda}} : \mathbb{R}^D \to \mathbb{R}^L$ and $f_{\boldsymbol{\psi}} : \mathbb{R}^L \to \mathbb{R}^D$ are fully connected feedforward neural networks with learnable parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\psi}$. The output layers of $g_{\boldsymbol{\lambda}}$ and $f_{\boldsymbol{\psi}}$ have no (i.e. have linear) activation functions. $\mathcal{N}$ denotes the normal distribution, $\boldsymbol{I}_N$ is the $N \times N$ identity matrix, and $\mathbf{0}_N$ is the vector of all zeros of length N.

For each of the architectures below, explain whether it's **necessary** to use the reparametrization trick to compute the gradient of the loss $\mathcal{L}$ w.r.t. **both** $\boldsymbol{\lambda}$ and $\boldsymbol{\psi}$. Answer "Yes" or "No" and provide a justification. If the answer is "Yes", modify the code to implement the reparametrization trick.

a) Model 1

$$
\begin{aligned}
\boldsymbol{z}_i &\sim \mathcal{N}(\boldsymbol{x}_i, \boldsymbol{I}_D) \\
\boldsymbol{h}_i &= g_{\boldsymbol{\lambda}}(\boldsymbol{z}_i) \\
\tilde{\boldsymbol{x}}_i &= f_{\boldsymbol{\psi}}(\boldsymbol{h}_i) \\
\mathcal{L} &= \|\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i\|_2^2
\end{aligned}
$$

No, the samples do not depend on **learnable** parameters here.

b) Model 2

$$
\begin{aligned}
\boldsymbol{h}_i &= g_{\boldsymbol{\lambda}}(\boldsymbol{x}_i) \\
\boldsymbol{z}_i &\sim \mathcal{N}(\boldsymbol{h}_i, \boldsymbol{I}_L) \\
\tilde{\boldsymbol{x}}_i &= f_{\boldsymbol{\psi}}(\boldsymbol{z}_i) \\
\mathcal{L} &= \|\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i\|_2^2
\end{aligned}
$$

Yes, We need to replace line 2 with the following operations:

$$
\begin{aligned}
\boldsymbol{\epsilon}_i &\sim \mathcal{N}(\mathbf{0}_L, \boldsymbol{I}_L) \\
\boldsymbol{z}_i &= \boldsymbol{h}_i + \boldsymbol{\epsilon}_i
\end{aligned}
$$

c) Model 3

$$
\begin{aligned}
\boldsymbol{h}_i &= g_{\boldsymbol{\lambda}}(\boldsymbol{x}_i) \\
\boldsymbol{z}_i &\sim \mathcal{N}(\mathbf{0}_L, \boldsymbol{I}_L) \\
\tilde{\boldsymbol{x}}_i &= f_{\boldsymbol{\psi}}(\boldsymbol{h}_i + \boldsymbol{z}_i) \\
\mathcal{L} &= \|\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i\|_2^2
\end{aligned}
$$

No, the samples do not depend on **learnable** parameters here.

Problem 2: Consider the same setup as in the previous problem. The model specified below is **not well defined**. Your task is to find the problem with the model and modify the pseudo code to fix it.

In addition, if you think it's **necessary** to use the reparametrization trick, include it in your implementation.

$$\begin{aligned}
\boldsymbol{h}_i &= g_{\boldsymbol{\lambda}}(\boldsymbol{x}_i) \\
\boldsymbol{z}_i &\sim \mathcal{N}(\boldsymbol{0}_L, \operatorname{diag}(\boldsymbol{h}_i)) \\
\tilde{\boldsymbol{x}}_i &= f_{\boldsymbol{\psi}}(\boldsymbol{z}_i) \\
\mathcal{L} &= \|\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i\|_2^2
\end{aligned}$$

The covariance matrix of the normal distribution must be positive definite for the model to be well-defined. For the diagonal covariance, this means that all entries must be positive. We can ensure this by using the exponential function

$$\boldsymbol{h}_i = \exp(g_{\boldsymbol{\lambda}}(\boldsymbol{x}_i))$$

Since the samples $\boldsymbol{z}_i$ depend on the parameters $\boldsymbol{\lambda}$ of the encoder, we need to use the reparametrization trick when sampling $\boldsymbol{z}_i$. For this, we replace line 2 with the following operations:

$$\begin{aligned}
\boldsymbol{\epsilon}_i &\sim \mathcal{N}(\boldsymbol{0}_L, \boldsymbol{I}_L) \\
\boldsymbol{z}_i &= \boldsymbol{\epsilon}_i \odot \sqrt{\boldsymbol{h}_i}
\end{aligned}$$

Problem 3: The loss used in generative adversarial networks (GANs) can be written in the following form:

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} \mathbb{E}_{p^*(\boldsymbol{x})}[\log D_{\boldsymbol{\phi}}(\boldsymbol{x})] + \mathbb{E}_{p(\boldsymbol{z})}[\log(1 - D_{\boldsymbol{\phi}}(f_{\boldsymbol{\theta}}(\boldsymbol{z})))]$$

where $p^*(\boldsymbol{x})$ is the true data distribution, $p(\boldsymbol{z})$ is the distribution of the noise, $f_{\boldsymbol{\theta}}$ is the generator, and $D_{\boldsymbol{\phi}}$ is the discriminator.

a) For a given generator (fixed parameters $\boldsymbol{\theta}$) assume there exists a discriminator $D_{\boldsymbol{\phi}^*}(\boldsymbol{x})$ with parameters $\boldsymbol{\phi}^*$ such that for all $\boldsymbol{x}$:

$$D_{\boldsymbol{\phi}^*}(\boldsymbol{x}) = \frac{p^*(\boldsymbol{x})}{p^*(\boldsymbol{x}) + p_{\boldsymbol{\theta}}(\boldsymbol{x})}$$

where $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ is the distribution learned by the generator. Show that $D_{\boldsymbol{\phi}^*}$ is **optimal**, i.e. $\boldsymbol{\phi}^* = \arg\max_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$.

Hint: $\arg\max_y[a \log(y) + b \log(1 - y)] = \frac{a}{a+b}$ for any $a, b \in \mathbb{R}_0^+, a + b > 0$.

$$\max_{D_{\boldsymbol{\phi}}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \max_{D_{\boldsymbol{\phi}}} \; \mathbb{E}_{p^*(\boldsymbol{x})}[\log D_{\boldsymbol{\phi}}(\boldsymbol{x})] + \mathbb{E}_{p(\boldsymbol{z})}[\log(1 - D_{\boldsymbol{\phi}}(f_{\boldsymbol{\theta}}(\boldsymbol{z})))] \tag{1}$$

$$= \max_{D_{\boldsymbol{\phi}}} \; \mathbb{E}_{p^*(\boldsymbol{x})}[\log D_{\boldsymbol{\phi}}(\boldsymbol{x})] + \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{x})}[\log(1 - D_{\boldsymbol{\phi}}(\boldsymbol{x}))] \tag{2}$$

$$= \max_{D_{\boldsymbol{\phi}}} \int [p^*(\boldsymbol{x}) \log D_{\boldsymbol{\phi}}(\boldsymbol{x}) + p_{\boldsymbol{\theta}}(\boldsymbol{x}) \log(1 - D_{\boldsymbol{\phi}}(\boldsymbol{x}))] \, d\boldsymbol{x} \tag{3}$$

It's not obvious how we can find the discriminator $D_{\boldsymbol{\phi}^*}$ that maximizes this integral. What if instead of maximizing the integral, we found the optimal discriminator **for every single $\boldsymbol{x}$**? This would only allow us to achieve higher values of our optimization objective

$$\leq \int \max_{D_{\boldsymbol{\phi}}} [p^*(\boldsymbol{x}) \log D_{\boldsymbol{\phi}}(\boldsymbol{x}) + p_{\boldsymbol{\theta}}(\boldsymbol{x}) \log(1 - D_{\boldsymbol{\phi}}(\boldsymbol{x}))] \, d\boldsymbol{x} \tag{4}$$

Finding the optimal discriminator for specific fixed $\boldsymbol{x}$ is easy. We use the formula from the problem statement, where we set $y = D_{\boldsymbol{\phi}}(\boldsymbol{x})$, $a = p^*(\boldsymbol{x})$, and $b = p_{\boldsymbol{\theta}}(\boldsymbol{x})$.

$$\begin{aligned} D_{\boldsymbol{\phi}^*}(\boldsymbol{x}) &= \arg\max_{D_{\boldsymbol{\phi}}(\boldsymbol{x})} (p^*(\boldsymbol{x}) \log D_{\boldsymbol{\phi}}(\boldsymbol{x}) + p_{\boldsymbol{\theta}}(\boldsymbol{x}) \log(1 - D_{\boldsymbol{\phi}}(\boldsymbol{x}))) \\ &= \frac{p^*(\boldsymbol{x})}{p^*(\boldsymbol{x}) + p_{\boldsymbol{\theta}}(\boldsymbol{x})} \end{aligned} \tag{5}$$

That is, if our discriminator satisfies the property from Equation 5 for every $\boldsymbol{x}$, then it maximizes the expression under the integral sign in Equation 4 for every $\boldsymbol{x}$. Finally, if our discriminator $D_{\boldsymbol{\phi}^*}$ maximizes the expression under the integral sign in Equation 4 for every $\boldsymbol{x}$, then it also maximizes the entire integral in Equation 3.

Putting everything together, we have shown, that the discriminator that satisfies Equation 5 for every point $\boldsymbol{x}$ is the optimal discriminator for the GAN loss (assuming fixed generator parameters $\boldsymbol{\theta}$).

b) What is the value of the optimal $D_{\boldsymbol{\phi}^*}(\boldsymbol{x})$ when:

(i) The generator is optimal i.e. $p_{\boldsymbol{\theta}}(\boldsymbol{x}) = p^*(\boldsymbol{x})$

(ii) The generator assigns a zero probability $p_{\boldsymbol{\theta}}(\boldsymbol{x}) = 0$ to a sample $\boldsymbol{x}$ whereas $p^*(\boldsymbol{x}) \neq 0$

(iii) The generator assigns a non-zero probability $p_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq 0$ to a sample $\boldsymbol{x}$ whereas $p^*(\boldsymbol{x}) = 0$

- If $p_{\boldsymbol{\theta}}(\boldsymbol{x}) = p^*(\boldsymbol{x})$, then $D_{\boldsymbol{\phi}^*}(\boldsymbol{x}) = \frac{1}{2}$
- If $p_{\boldsymbol{\theta}}(\boldsymbol{x}) = 0$ and $p^*(\boldsymbol{x}) \neq 0$, then $D_{\boldsymbol{\phi}^*}(\boldsymbol{x}) = 1$. The discriminator classifies all such samples as “Real”.
- If $p_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq 0$ and $p^*(\boldsymbol{x}) = 0$, then $D_{\boldsymbol{\phi}^*}(\boldsymbol{x}) = 0$. The discriminator classifies all such samples as “Fake”.