Deep Learning, Homework 5

Javad Razi **401204354**

Problem One

(a) To construct two distributions for which JSD is not differentiable with respect to parameters P and Q, we can consider the following distributions:

- P is a discrete distribution with two points: $(0, 0.5)$ and $(1, 0.5)$.
- Q is a continuous distribution with a density function $q(x) = \frac{1}{2}\exp(-|x|)$.

The JSD between these two distributions is given by:

$$JSD(P, Q) = \frac{1}{2}D_{KL}(P \parallel \frac{P+Q}{2}) + \frac{1}{2}D_{KL}(Q \parallel \frac{P+Q}{2})$$

$$= \frac{1}{2}\left[0.5 \log \frac{0.5}{0.75} + 0.5 \log \frac{0.5}{0.25}\right] + \frac{1}{2}\int_{-\infty}^{\infty} \frac{1}{2}e^{-|x|} \log \frac{e^{-|x|}}{0.75e^{-|x|} + 0.25} dx$$

$$= \frac{1}{2}[-0.2231 + 0.2231] + \frac{1}{2}\int_{-\infty}^{\infty} \frac{1}{2}e^{-|x|} \log \frac{1}{0.75 + 0.25e^{|x|}} dx$$

$$= \frac{1}{2}\int_{-\infty}^{\infty} \frac{1}{2}e^{-|x|} \log \frac{1}{0.75 + 0.25e^{|x|}} dx$$

The integral in the above expression does not have a closed-form solution, and it is not differentiable with respect to the parameters of P and Q. Therefore, JSD is not differentiable with respect to the parameters of P and Q for these two distributions.

(b) The objective functions L_D and L_G are used for training GAN networks. The discriminator loss L_D encourages the discriminator to correctly classify real and fake images, while the generator loss L_G encourages the generator to produce fake images that are indistinguishable from real images.

The two-step optimization process involves updating the discriminator and generator parameters in opposite directions. In the first step, the discriminator parameters are updated in the direction opposite to the gradient of L_D. This is done to maximize the discriminator's ability to distinguish between real and fake images. In the second step, the generator parameters are updated in the direction of the gradient of L_G. This is done to minimize the generator's loss and encourage it to produce more realistic fake images.

The optimization process continues until the discriminator and generator reach an equilibrium, where the discriminator is unable to distinguish between real and fake images.

At this point, the generator is able to produce realistic fake images that are indistinguishable from real images.

(2) When using pretrained backbones in the discriminator of GAN networks, these networks can easily identify fake images. This good performance can lead to a vanishing gradient problem.

To show this, consider the following:

- The discriminator's output for fake images tends towards zero: $D_\sigma(G_\theta(z)) \approx 0$.
- The network's output uses a sigmoid activation function: $\sigma(x) = \frac{1}{1+e^{-x}}$.

The gradient of the sigmoid function is given by:

$$\frac{d\sigma(x)}{dx} = \sigma(x)(1 - \sigma(x))$$

Substituting $x = h_\phi(G_\theta(z))$ into the above equation, we get:

$$\frac{d\sigma(h_\phi(G_\theta(z)))}{dh_\phi(G_\theta(z))} = \sigma(h_\phi(G_\theta(z)))(1 - \sigma(h_\phi(G_\theta(z))))$$

Since $D_\sigma(G_\theta(z)) \approx 0$, we have $\sigma(h_\phi(G_\theta(z))) \approx 0$. Therefore, the gradient of the sigmoid function becomes:

$$\frac{d\sigma(h_\phi(G_\theta(z)))}{dh_\phi(G_\theta(z))} \approx 0$$

This means that the gradient of the discriminator's output with respect to the generator's parameters is approximately zero. As a result, the generator will not be able to learn to produce more realistic fake images, and the network will suffer from vanishing gradient.

(3) To show that L_D is minimized when $D^* = D$, where:

$$D^*(x) = \frac{P_{\text{data}}(x)}{P_{\text{data}}(x) + P_\theta(x)}$$

we can use the following steps:

1. Substitute D^* into L_D:

$$L_D(D^*, \theta) = \mathbb{E}_{x \sim P_{\text{data}}}[\log D^*(x)] - \mathbb{E}_{z \sim \mathcal{N}(0,1)}[\log(1 - D^*(G_\theta(z)))]$$

2. Expand the logarithms:

$$L_D(D^*, \theta) = \mathbb{E}_{x \sim P_{\text{data}}}\left[\log \frac{P_{\text{data}}(x)}{P_{\text{data}}(x) + P_\theta(x)}\right] - \mathbb{E}_{z \sim \mathcal{N}(0,1)}\left[\log\left(1 - \frac{P_{\text{data}}(G_\theta(z))}{P_{\text{data}}(G_\theta(z)) + P_\theta(G_\theta(z))}\right)\right]$$

3. Simplify the expressions inside the expectations:

$$L_D(D^*, \theta) = \mathbb{E}_{x \sim P_{\text{data}}}\left[\log P_{\text{data}}(x) - \log(P_{\text{data}}(x) + P_\theta(x))\right] - \mathbb{E}_{z \sim \mathcal{N}(0,1)}\left[\log(P_{\text{data}}(G_\theta(z)) + P_\theta(G_\theta(z)))\right.$$

4. Combine the two expectations into a single expectation:

$$L_D(D^*, \theta) = \mathbb{E}_{x \sim P_{\text{data}}} \left[\log \frac{P_{\text{data}}(x)}{P_{\text{data}}(x) + P_\theta(x)} \right] + \mathbb{E}_{x \sim P_\theta} \left[\log \frac{P_{\text{data}}(x)}{P_{\text{data}}(x) + P_\theta(x)} \right]$$

5. Recognize that the two expectations are equal to the Jensen-Shannon Divergence between P_{data} and P_θ:

$$L_D(D^*, \theta) = -2 \cdot JSD(P_{\text{data}} \parallel P_\theta)$$

6. Since the Jensen-Shannon Divergence is always non-negative, $L_D(D^*, \theta)$ is minimized when $JSD(P_{\text{data}} \parallel P_\theta)$ is minimized. This occurs when $D^* = D$.

(4) The ideal generator in a GAN network produces fake data derived from the training data distribution P_{data}. As seen in the previous section, the network aims to minimize the distance between the generator's distribution and the training data distribution using the Jensen-Shannon Divergence.

However, there are several drawbacks to using JSD in the GAN model:

- **Non-convexity:** JSD is a non-convex function, which means that it can have multiple local minima. This can make it difficult to train the GAN network to find the global minimum, which corresponds to the ideal generator.
- **Asymmetry:** JSD is not symmetric, which means that $JSD(P_{\text{data}} \parallel P_\theta)$ is not equal to $JSD(P_\theta \parallel P_{\text{data}})$. This can make it difficult to interpret the results of the GAN training process.
- **Computational cost:** JSD is computationally expensive to calculate, especially for high-dimensional data. This can make it difficult to train GAN networks on large datasets.

(5) The Wasserstein GAN paper introduces the Earth Mover's Distance (EMD) as a metric for GAN networks. EMD is preferable over KL Divergence and JS Divergence for the following reasons:

- **Convexity:** EMD is a convex function, which means that it has a unique global minimum. This makes it easier to train the GAN network to find the ideal generator.
- **Symmetry:** EMD is symmetric, which means that $W(P_{\text{data}}, P_\theta)$ is equal to $W(P_\theta, P_{\text{data}})$. This makes it easier to interpret the results of the GAN training process.
- **Computational efficiency:** EMD can be calculated efficiently using linear programming techniques. This makes it possible to train GAN networks on large datasets.

Problem Two

(a) To show that $F(q) = \log p(x) - D_{KL}(q(z) \parallel p(z|x))$, we can use the following steps:

1. Start with the definition of the variational free energy:

$$F(q) = \mathbb{E}_q[\log p(x|z)] - D_{KL}(q(z) \parallel p(z))$$

2. Add and subtract $\log p(x)$ to the first term:

$$F(q) = \log p(x) + \mathbb{E}_q[\log p(x|z) - \log p(x)] - D_{KL}(q(z) \parallel p(z))$$

3. Recognize that $\mathbb{E}_q[\log p(x|z) - \log p(x)] = \log p(x) - \log p(x) = 0$. Therefore, we can simplify the equation to:

$$F(q) = \log p(x) - D_{KL}(q(z) \parallel p(z))$$

4. Finally, use the fact that $p(z|x) = \frac{p(x,z)}{p(x)}$ to rewrite the KL divergence term:

$$D_{KL}(q(z) \parallel p(z)) = \mathbb{E}_q[\log q(z) - \log p(z)] = \mathbb{E}_q[\log q(z) - \log p(x,z) + \log p(x)]$$
$$= \mathbb{E}_q[\log q(z) - \log p(x,z)] + \log p(x) = D_{KL}(q(z) \parallel p(z|x)) + \log p(x)$$

5. Substituting this expression back into the equation for $F(q)$, we get:

$$F(q) = \log p(x) - (D_{KL}(q(z) \parallel p(z|x)) + \log p(x)) = \log p(x) - D_{KL}(q(z) \parallel p(z|x))$$

Therefore, we have shown that $F(q) = \log p(x) - D_{KL}(q(z) \parallel p(z|x))$.

(b) To provide an explicit formula for $D_{KL}(q_i(z_i) \parallel p_i(z_i))$ in terms of μ_i and σ_i^2, we can use the following steps:

1. Start with the definition of the KL divergence:

$$D_{KL}(q_i(z_i) \parallel p_i(z_i)) = \mathbb{E}_{q_i(z_i)}[\log q_i(z_i) - \log p_i(z_i)]$$

2. Substitute the expressions for $q_i(z_i)$ and $p_i(z_i)$:

$$D_{KL}(q_i(z_i) \parallel p_i(z_i)) = \mathbb{E}_{q_i(z_i)}\left[\log \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(z_i - \mu_i)^2}{2\sigma_i^2}\right) - \log \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z_i^2}{2}\right)\right]$$

3. Simplify the expression:

$$D_{KL}(q_i(z_i) \parallel p_i(z_i)) = \mathbb{E}_{q_i(z_i)}\left[\log \frac{1}{\sqrt{2\pi\sigma_i^2}} - \frac{(z_i - \mu_i)^2}{2\sigma_i^2} - \log \frac{1}{\sqrt{2\pi}} + \frac{z_i^2}{2}\right]$$

4. Expand the expectation:

$$D_{KL}(q_i(z_i) \parallel p_i(z_i)) = \log \frac{1}{\sqrt{2\pi\sigma_i^2}} - \frac{1}{2\sigma_i^2}\mathbb{E}_{q_i(z_i)}[(z_i - \mu_i)^2] - \log \frac{1}{\sqrt{2\pi}} + \frac{1}{2}\mathbb{E}_{q_i(z_i)}[z_i^2]$$

5. Recognize that $\mathbb{E}_{q_i(z_i)}[(z_i - \mu_i)^2] = \sigma_i^2$ and $\mathbb{E}_{q_i(z_i)}[z_i^2] = \mu_i^2 + \sigma_i^2$. Therefore, we can simplify the expression to:

$$D_{KL}(q_i(z_i) \parallel p_i(z_i)) = \frac{1}{2}\left[\log\frac{\sigma_i^2}{1} + \frac{1}{\sigma_i^2}(\sigma_i^2 + \mu_i^2 - \mu_i^2) - 1\right]$$

$$= \frac{1}{2}\left[\log\sigma_i^2 + 1 - 1\right] = \frac{1}{2}\log\sigma_i^2$$

Therefore, we have shown that $D_{KL}(q_i(z_i) \parallel p_i(z_i)) = \frac{1}{2}\log\sigma_i^2$.

(c) To derive that $D_{KL}(q(z) \parallel p(z)) = \sum_i D_{KL}(q_i(z_i) \parallel p_i(z_i))$, we can use the following steps:

1. Start with the definition of the KL divergence:

$$D_{KL}(q(z) \parallel p(z)) = \mathbb{E}_q[\log q(z) - \log p(z)]$$

2. Substitute the expressions for $q(z)$ and $p(z)$:

$$D_{KL}(q(z) \parallel p(z)) = \mathbb{E}_q\left[\log\frac{1}{\sqrt{(2\pi)^D|\Sigma|}}\exp\left(-\frac{1}{2}(z-\mu)^\top\Sigma^{-1}(z-\mu)\right) - \log\frac{1}{\sqrt{(2\pi)^D}}\exp\left(-\frac{1}{2}z^\top z\right)\right]$$

3. Simplify the expression:

$$D_{KL}(q(z) \parallel p(z)) = \mathbb{E}_q\left[\log\frac{1}{\sqrt{(2\pi)^D|\Sigma|}} - \frac{1}{2}(z-\mu)^\top\Sigma^{-1}(z-\mu) - \log\frac{1}{\sqrt{(2\pi)^D}} + \frac{1}{2}z^\top z\right]$$

4. Expand the expectation:

$$D_{KL}(q(z) \parallel p(z)) = \log\frac{1}{\sqrt{(2\pi)^D|\Sigma|}} - \frac{1}{2}\mathbb{E}_q[(z-\mu)^\top\Sigma^{-1}(z-\mu)] + \frac{1}{2}\mathbb{E}_q[z^\top z] - \log\frac{1}{\sqrt{(2\pi)^D}}$$

5. Recognize that $\mathbb{E}_q[(z-\mu)^\top\Sigma^{-1}(z-\mu)] = \mathrm{tr}(\Sigma^{-1}\Sigma) = D$ and $\mathbb{E}_q[z^\top z] = \mathrm{tr}(\Sigma) + \mu^\top\mu = D + \mu^\top\mu$. Therefore, we can simplify the expression to:

$$D_{KL}(q(z) \parallel p(z)) = \frac{1}{2}\left[\log\frac{|\Sigma|}{1} + D - D - \mu^\top\mu - 1\right] = \frac{1}{2}\left[\log|\Sigma| - \mu^\top\mu - 1\right]$$

6. Since $q(z)$ is a diagonal Gaussian distribution, we have $|\Sigma| = \prod_i \sigma_i^2$ and $\mu^\top\mu = \sum_i \mu_i^2$. Therefore, we can write:

$$D_{KL}(q(z) \parallel p(z)) = \frac{1}{2}\left[\sum_i \log\sigma_i^2 - \sum_i \mu_i^2 - 1\right] = \sum_i \frac{1}{2}\left[\log\sigma_i^2 - \mu_i^2 - 1\right]$$

7. Finally, we can recognize that $D_{KL}(q_i(z_i) \parallel p_i(z_i)) = \frac{1}{2}\log\sigma_i^2$ and $\mu_i^2 = \mathbb{E}_{q_i(z_i)}[z_i^2] - \mathbb{E}_{q_i(z_i)}[z_i]^2$. Therefore, we have shown that:

$$D_{KL}(q(z) \parallel p(z)) = \sum_i D_{KL}(q_i(z_i) \parallel p_i(z_i))$$

Problem Three

(a) The main difference between VAE and β-VAE is in the way they regularize the latent representation z.

- **VAE:** In VAE, the latent representation z is regularized by minimizing the KL divergence between the approximate posterior distribution $q_\phi(z|x)$ and the prior distribution $p(z)$. This encourages the latent representation to be close to the prior distribution, which can lead to a loss of information.
- **β-VAE:** In β-VAE, the latent representation z is regularized by adding a KL divergence term to the loss function, with a weight factor β. This encourages the latent representation to be close to the prior distribution, but it also allows for more flexibility in the representation. This can help to improve the disentanglement of the latent factors, which means that each latent factor is responsible for a different aspect of the data.

(b) The optimization expression for β-VAE is:

$$\max_\theta \mathbb{E}_{x\sim D}\left[\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]\right] \text{ subject to } D_{KL}(q_\phi(z|x) \parallel p(z)) < \varepsilon$$

This expression means that we want to maximize the expected log likelihood of the data x given the latent variable z, while also constraining the KL divergence between the approximate posterior distribution $q_\phi(z|x)$ and the prior distribution $p(z)$ to be less than some threshold ε.

The intuition behind this optimization expression is that we want to find a latent representation z that is both informative (i.e., it helps to reconstruct the data x) and disentangled (i.e., it captures the different aspects of the data). The KL divergence term encourages the latent representation to be close to the prior distribution, which helps to disentangle the latent factors. The log likelihood term encourages the latent representation to be informative, as it measures how well the data can be reconstructed from the latent representation.

(c) The cost function for β-VAE is derived from the optimization expression as follows:

$$\mathcal{L}(\theta, \phi) = -\mathbb{E}_{x\sim D}\left[\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]\right] + \beta D_{KL}(q_\phi(z|x) \parallel p(z))$$

This cost function is simply the negative of the optimization expression, plus the KL divergence term. The negative sign is added so that we can minimize the cost function.

(d) The disentanglement metric measures how well the latent representation z captures the different aspects of the data. It is defined as follows:

$$D = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{d}\sum_{j=1}^{d} I(z_j; x_i)$$

Where N is the number of data points, d is the dimensionality of the latent representation, z_j is the j-th latent dimension, x_i is the i-th data point, and $I(z_j; x_i)$ is the mutual information between z_j and x_i.

The disentanglement metric is important because it measures how well the latent representation is able to capture the different aspects of the data. A high disentanglement metric indicates that the latent representation is able to disentangle the different factors of variation in the data, which can be useful for tasks such as image generation and manipulation.

Changing β affects the network training in the following ways:

- **High β:** A high β value will encourage the latent representation to be close to the prior distribution. This can help to improve the disentanglement of the latent factors, but it can also lead to a loss of information.
- **Low β:** A low β value will allow for more flexibility in the latent representation. This can help to improve the reconstruction quality, but it can also lead to a less disentangled latent representation.

The optimal value of β will depend on the specific dataset and task.

Problem Four

Part One

(a) To express $p_i(z_i)$ in terms of z_{i-1}, we can start with Equation 4:

$$p_i(z_i) = p_{i-1}(z_{i-1}) |\det \frac{df_{i-1}}{dz_{i-1}}|$$

Taking the logarithm of both sides, we get:

$$\log p_i(z_i) = \log p_{i-1}(z_{i-1}) + \log |\det \frac{df_{i-1}}{dz_{i-1}}|$$

Using the inverse function theorem, we know that:

$$\frac{dz_{i-1}}{dz_i} = \frac{1}{df_i/dz_{i-1}}$$

Taking the determinant of both sides, we get:

$$\det \frac{dz_{i-1}}{dz_i} = \frac{1}{\det \frac{df_i}{dz_{i-1}}}$$

Substituting this into the above equation, we get:

$$\log p_i(z_i) = \log p_{i-1}(z_{i-1}) - \log |\det \frac{dz_{i-1}}{dz_i}|$$

Since $z_0 = x$, we can rewrite the above equation as:

$$\log p_i(z_i) = \log p_{i-1}(z_{i-1}) - \log |\det \frac{df_i}{dz_{i-1}}|$$

(b) To show that $\log p(x)$ is equivalent to the given expression, we can start with the equation we derived in part (α):

$$\log p_i(z_i) = \log p_{i-1}(z_{i-1}) - \log |\det \frac{df_i}{dz_{i-1}}|$$

Summing both sides from $i = 1$ to K, we get:

$$\sum_{i=1}^{K} \log p_i(z_i) = \sum_{i=1}^{K} \log p_{i-1}(z_{i-1}) - \sum_{i=1}^{K} \log |\det \frac{df_i}{dz_{i-1}}|$$

Substituting $z_0 = x$ and $p_K(z_K) = p(x)$, we get:

$$\log p(x) = \log p_0(z_0) - \sum_{i=1}^{K} \log |\det \frac{df_i}{dz_{i-1}}|$$

Therefore, we have shown that $\log p(x)$ is equivalent to the given expression.

Part Two

(a) To demonstrate that the transformation is invertible, we need to show that for any y, we can find a unique x such that $f(x) = y$.

Let's consider the two parts of the transformation separately:

- For the first d dimensions, we have:

$$y_{1:d} = x_{1:d}$$

This is a simple linear transformation, and its inverse is simply:

$$x_{1:d} = y_{1:d}$$

- For the remaining dimensions $d+1$ to D, we have:

$$y_{d+1:D} = x_{d+1:D} \odot \exp(s(x_{1:d})) + t(x_{1:d})$$

To find the inverse of this transformation, we can solve for $x_{d+1:D}$:

$$x_{d+1:D} = \frac{y_{d+1:D} - t(x_{1:d})}{\exp(s(x_{1:d}))}$$

Since $s(\cdot)$ and $t(\cdot)$ are known functions, we can compute $x_{d+1:D}$ for any given $y_{d+1:D}$ and $x_{1:d}$.

Therefore, we have shown that the transformation is invertible.

(b) To show that the determinant of the Jacobian is computable, we can use the following formula:

$$\det(J_f) = \prod_{i=1}^{D} \frac{\partial y_i}{\partial x_i}$$

where J_f is the Jacobian matrix of the transformation f.

For the first d dimensions, we have:

$$\frac{\partial y_i}{\partial x_i} = 1$$

for $i = 1, \ldots, d$.

For the remaining dimensions $d+1$ to D, we have:

$$\frac{\partial y_i}{\partial x_i} = \exp(s(x_{1:d}))$$

for $i = d+1, \ldots, D$.

Therefore, the determinant of the Jacobian is:

$$\begin{aligned}\det(J_f) &= \prod_{i=1}^{d} 1 \cdot \prod_{i=d+1}^{D} \exp(s(x_{1:d})) \\ &= \exp\left(\sum_{i=d+1}^{D} s(x_{1:d})\right)\end{aligned}$$

Since $s(\cdot)$ is a known function, we can compute the determinant of the Jacobian for any given x.

Hence, we have shown that the determinant of the Jacobian is computable.

Problem Five

(a) Inception Score (IS):

- IS measures the quality and diversity of generated images by calculating the KL divergence between the predicted labels of generated images and the distribution of labels for real images.
- A higher IS score indicates better image quality and diversity.
- IS is easy to compute and interpret.

Fréchet Inception Distance (FID):

- FID measures the similarity between the distributions of real and generated images by calculating the Fréchet distance between the activations of the penultimate layer of an Inception network.
- A lower FID score indicates better image quality and diversity.
- FID is more computationally expensive to compute than IS, but it is generally considered to be a more reliable metric.

(b) Shortcomings of FID:

- FID can be sensitive to outliers in the real image distribution. For example, if the real image distribution contains a small number of images that are very different from the rest of the images, FID can be high even if the generated images are of good quality.
- FID does not measure the diversity of generated images directly. It is possible for a generative model to achieve a low FID score by generating a small number of high-quality images, even if the model does not generate a diverse set of images.

Example: consider a generative model that generates only images of cats. The FID score of this model would be low, because the distribution of generated images would be very similar to the distribution of real images of cats. However, the model would not be considered to be diverse, because it only generates images of cats.

(c) Density and Convergence:

- **Precision:** Precision measures the fraction of generated images that are classified as real by a classifier trained on real images.
- **Recall:** Recall measures the fraction of real images that are classified as real by a classifier trained on generated images.
- **Addressing the problem of 'real outlier' in precision:** The article addresses the problem of real outliers by using a nearest-neighbor classifier to identify and remove outliers from the real image distribution. This helps to ensure that the precision metric is not affected by a small number of real images that are very different from the rest of the images.
- **Addressing the issue of 'unrealistic diverse samples' in recall:** The article addresses the issue of unrealistic diverse samples by using a two-stage evaluation process. In the first stage, a classifier is trained on real images. In the second stage, a classifier is trained on generated images. The recall metric is then calculated by measuring the fraction of real images that are classified as real by the classifier trained on generated images. This helps to ensure that the recall metric is not affected by generated images that are unrealistic or diverse.
