Hena Ghonia (20213256)

Question 1 (5-5-6)**.** Consider a latent variable model $p_\theta(\boldsymbol{x}) = \int p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})dz$, where $p(\boldsymbol{z}) = \mathcal{N}(\mathbf{0}, \boldsymbol{I}_K)$ and $\boldsymbol{z} \in \mathbb{R}^K$.The encoder network (aka "recognition model") of variational autoencoder, $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, is used to produce an approximate (variational) posterior distribution over latent variables $\boldsymbol{z}$ for any input datapoint $\boldsymbol{x}$.[1] This distribution is trained to match the true posterior by maximizing the evidence lower bound (ELBO):

$$\mathcal{L}(\theta, \phi; \boldsymbol{x}) = \mathbb{E}_{q_\phi}[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z})] - D_{\mathrm{KL}}(q_\phi(\boldsymbol{z} \mid \boldsymbol{x})||p(\boldsymbol{z}))$$

Let $\mathcal{Q}$ be the family of variational distributions with a feasible set of parameters $\mathcal{P}$; i.e. $\mathcal{Q} = \{q(\boldsymbol{z}; \pi) : \pi \in \mathcal{P}\}$; for example π can be mean and standard deviation of a normal distribution. We assume q_ϕ is parameterized by a neural network (with parameters ϕ) that outputs the parameters, $\pi_\phi(\boldsymbol{x})$, of the distribution $q \in \mathcal{Q}$, i.e. $q_\phi(\boldsymbol{z}|\boldsymbol{x}) := q(\boldsymbol{z}; \pi_\phi(\boldsymbol{x}))$.

1.1 Show that maximizing the expected complete data log likelihood (ECLL)

$$\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})]$$

for a fixed $q(\boldsymbol{z}|\boldsymbol{x})$, wrt the model parameter θ, is equivalent to maximizing

$$\log p_\theta(\boldsymbol{x}) - D_{\mathrm{KL}}(q(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x}))$$

This means the maximizer of the ECLL coincides with that of the marginal likelihood only if $q(\boldsymbol{z}|\boldsymbol{x})$ perfectly matches $p(\boldsymbol{z}|\boldsymbol{x})$.

Solution: As we know that Variational method are based on following relationship from slide no.17 from VAE.pdf(Lecture slides):

$$\log p_\theta(\boldsymbol{x}) = \mathcal{L}(q, p_\theta) + D_{KL}\left(q(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})\right) \tag{1}$$

Where $\mathcal{L}(q, p_\theta)$ is Evidence lower bound.
Defining variational lower bound on data likelihood as:

$$\mathcal{L}(q, p_\theta) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z}|\boldsymbol{x})\right]$$

where $\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}, \boldsymbol{z})\right]$ is given as expected complete data log likelihood.
Substituting $\mathcal{L}(q, p_\theta)$ in equation (1):

$$\log p_\theta(\boldsymbol{x}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z}|\boldsymbol{x})\right] + D_{KL}\left(q(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})\right)$$

$$\log p_\theta(\boldsymbol{x}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})\right] - \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log q(\boldsymbol{z}|\boldsymbol{x})\right] + D_{KL}\left(q(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})\right)$$

$$\log p_\theta(\boldsymbol{x}) - D_{KL}\left(q(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})\right) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})\right] - \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log q(\boldsymbol{z}|\boldsymbol{x})\right]$$

From above equation maximizing $\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})]$ when $q(\boldsymbol{z}|\boldsymbol{x})$ perfectly matches $p(z|x)$ is equivalent to maximizing $\log p_\theta(\boldsymbol{x}) - D_{KL}\left(q(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})\right)$.

1. Using a recognition model in this way is known as "amortized inference"; this can be contrasted with traditional variational inference approaches (see, e.g., Chapter 10 of Bishop's *Pattern Recognition an Machine Learning*), which fit a variational posterior independently for each new datapoint.

It can also be proved in following way. As we know that $q(\boldsymbol{z}|\boldsymbol{x})$ and $p(\boldsymbol{z})$ are independent of parameter θ, we can write in following form using the condition that $q(\boldsymbol{z}|\boldsymbol{x})$ perfectly matches $p(z|x)$:

$$\begin{aligned}
&\arg\max_{\theta} \left\{\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})]\right\} \\
&= \arg\max_{\theta} \left\{\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{p_\theta(\boldsymbol{x}|\boldsymbol{z})}\frac{p_\theta(\boldsymbol{x}|\boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{x})}\right]\right\} \\
&= \arg\max_{\theta} \left\{\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{x})}\right]\right\} \\
&= \arg\max_{\theta}\{\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z}) - \log q(\boldsymbol{z}|\boldsymbol{x})\right]\} \ [\because p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z}) = p_\theta(\boldsymbol{x},\boldsymbol{z})] \\
&= \arg\max_{\theta}\{\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x},\boldsymbol{z}) - \log q(\boldsymbol{z}|\boldsymbol{x})\right]\} \\
&= \arg\max_{\theta}\{\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{z}|\boldsymbol{x})p_\theta(\boldsymbol{x}) - \log q(\boldsymbol{z}|\boldsymbol{x})\right]\} \ [\because p_\theta(\boldsymbol{x},\boldsymbol{z}) = p_\theta(\boldsymbol{z}|\boldsymbol{x})p_\theta(\boldsymbol{x})] \\
&= \arg\max_{\theta}\{\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log q(\boldsymbol{z}|\boldsymbol{x})p_\theta(\boldsymbol{x}) - \log q(\boldsymbol{z}|\boldsymbol{x})\right]\} \ [\because q(\boldsymbol{x},\boldsymbol{z}) = p(\boldsymbol{x},\boldsymbol{z})] \\
&= \arg\max_{\theta}\{\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x})\right] - \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[q(\boldsymbol{z}|\boldsymbol{x}) - \log p_\theta(\boldsymbol{z}|\boldsymbol{x})\right]\} \\
&= \arg\max_{\theta}\{\log p_\theta(x) - \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{q(\boldsymbol{z}|\boldsymbol{x})}{p_\theta(\boldsymbol{z}|\boldsymbol{x})}\right]\} \\
&= \arg\max_{\theta}\{\log p_\theta(\boldsymbol{x}) - D_{KL}\left(q(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})\right)\}
\end{aligned} \tag{2}$$

1.2 Consider a finite training set $\{\boldsymbol{x}_i : i \in \{1, ..., n\}\}$, n being the size the training data. Let ϕ^* be the maximizer $\arg\max_\phi \sum_{i=1}^n \mathcal{L}(\theta, \phi; \boldsymbol{x}_i)$ with θ fixed. In addition, for each $\boldsymbol{x}_i$ let $q_i \in \mathcal{Q}$ be an "instance-dependent" variational distribution, and denote by q_i^* the maximizer of the corresponding ELBO. Compare $D_{\mathrm{KL}}(q_{\phi^*}(\boldsymbol{z}|\boldsymbol{x}_i)||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i))$ and $D_{\mathrm{KL}}(q_i^*(\boldsymbol{z})||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i))$. Which one is bigger ?

Solution: Inference gap of q^* can be written as from paper - 'Inference Suboptimality in Variational Autoencoders':

$$\log p_\theta(\boldsymbol{x}) - \mathcal{L}\left[q_{\phi^*}(\boldsymbol{z}|\boldsymbol{x})\right] = \log p_\theta(\boldsymbol{x}_i) - \mathcal{L}\left[q_i^*(\boldsymbol{z})\right] + \mathcal{L}\left[q_i^*(\boldsymbol{z})\right] - \mathcal{L}\left[q_{\phi^*}(\boldsymbol{z}|\boldsymbol{x}_i)\right]$$

where $\log p_\theta(\boldsymbol{x}_i) - \mathcal{L}\left[q_i^*(\boldsymbol{z})\right]$ is Approximation gap and $\mathcal{L}\left[q_i^*(\boldsymbol{z})\right] - \mathcal{L}\left[q_{\phi^*}(\boldsymbol{z}|\boldsymbol{x}_i)\right]$ is Amortization gap. As we know that Variational method are based on following relationship from slide no.17 from VAE.pdf(Lecture slides):

$$\log p_\theta(\boldsymbol{x}) - D_{KL}\left(q(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})\right) = \mathcal{L}\left[q(\boldsymbol{z}|\boldsymbol{x})\right]]$$

$$\log p_\theta(\boldsymbol{x}) - \mathcal{L}\left[q(\boldsymbol{z}|\boldsymbol{x})\right]] = D_{KL}\left(q(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})\right)$$

Replacing above form of formula in inference gap formula for $\log p_\theta(\boldsymbol{x}) - \mathcal{L}\left[q_{\phi^*}(\boldsymbol{z}|\boldsymbol{x})\right]$ and $\log p_\theta(\boldsymbol{x}_i) - \mathcal{L}\left[q_i^*(\boldsymbol{z})\right]$.

$$\log p_\theta(x) - \mathcal{L}\left[q_{\phi^*}(\boldsymbol{z}|\boldsymbol{x})\right] = D_{KL}\left(q_{\phi^*}(\boldsymbol{z}|\boldsymbol{x}_i)||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i)\right)$$

$$\log p_\theta(x_i) - \mathcal{L}\left[q_i^*(\boldsymbol{z})\right] = D_{KL}\left(q^*(\boldsymbol{z})||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i)\right)$$

Replacing above equations in inference gap formula:

$$D_{KL}\left(q_{\phi^*}(\boldsymbol{z}|\boldsymbol{x}_i)||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i)\right) = D_{KL}\left(q^*(\boldsymbol{z})||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i)\right) + \left(\mathcal{L}\left[q_i^*(\boldsymbol{z})\right] - \mathcal{L}\left[q_{\phi^*}(\boldsymbol{z}|\boldsymbol{x}_i)\right]\right)$$

As mentioned in question that q_i^* is maximizer of the corresponding ELBO i.e from $\mathcal{Q}$, $\left(\mathcal{L}\left[q_i^*(\boldsymbol{z})\right] - \mathcal{L}\left[q_{\phi^*}(\boldsymbol{z}|\boldsymbol{x}_i)\right]\right)$ will be positive in above equation so following form can be written:

$$D_{KL}\left(q_{\phi^*}(\boldsymbol{z}|\boldsymbol{x}_i)||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i)\right) \geq D_{KL}\left(q^*(\boldsymbol{z})||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i)\right)$$

1.3 Following the previous question, compare the two approaches in the second subquestion.

(a) in terms of bias of estimating the marginal likelihood via the ELBO, in the best case scenario (i.e. when both approaches are optimal within the respective families)
Solution: There is bias since we use Variational Inference method (given in lecture slide-9 of VAE.pdf) which has KL divergence and expected data log likelihood is the term in ELBO that depends on θ.

(b) from the computational point of view (efficiency)
Solution: Approximating $q_{\phi^*}(\boldsymbol{z}|\boldsymbol{x}_i)$ is more computational efficient than $q_i(z)$ as we don't need to update q_i for each x_i in finite training set.

(c) in terms of memory (storage of parameters)
Solution: Approximating $q_i^*(\boldsymbol{z})$ needs more parameter space than $q_{\phi^*}(\boldsymbol{z}|\boldsymbol{x}_i)$ since $q_i^*(\boldsymbol{z})$ needs to store parameter at each ith iteration and hence has n times more parameter than amortized approximation.

Question 2 (5-5-5-5)**.** One way to enforce autoregressive conditioning is via masking the weight parameters.[2] Consider a two-hidden-layer convolutional neural network without kernel flipping, with kernel size 3×3 and padding size 1 on each border (so that an input feature map of size 5×5 is convolved into a 5×5 output). Define mask of type A and mask of type B as

$$(\boldsymbol{M}^A)_{::ij} := \begin{cases} 1 & \text{if } i < 2 \\ 1 & \text{if } i = 2 \text{ and } j < 2 \\ 0 & \text{elsewhere} \end{cases} \qquad (\boldsymbol{M}^B)_{::ij} := \begin{cases} 1 & \text{if } i < 2 \\ 1 & \text{if } i = 2 \text{ and } j \leq 2 \\ 0 & \text{elsewhere} \end{cases}$$

where the index starts from 1. Masking is achieved by multiplying the kernel with the binary mask (elementwise). Specify the receptive field of the output pixel that corresponds to the third row and the third column (index 33 of Figure 1 (Left)) in each of the following 4 cases:

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	44	45
51	52	53	54	55

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	44	45
51	52	53	54	55

FIGURE 1 – (Left) 5×5 convolutional feature map. (Right) Template answer.

2. An example of this is the use of masking in the Transformer architecture.

1. If we use $\boldsymbol{M}^A$ for the first layer and $\boldsymbol{M}^A$ for the second layer.
   Solution: Figure 2 (Left)) shows if we use $\boldsymbol{M}^A$ for the first layer and (Right) shows result after we use $\boldsymbol{M}^A$ for the first layer and $\boldsymbol{M}^A$ for the second layer.

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	44	45
51	52	53	54	55

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	44	45
51	52	53	54	55

FIGURE 2 – (Left) After using $\boldsymbol{M}^A$ for the first layer. (Right) After using $\boldsymbol{M}^A$ for the second layer.

2. If we use $\boldsymbol{M}^A$ for the first layer and $\boldsymbol{M}^B$ for the second layer.
   Solution: Figure 3 (Left)) shows if we use $\boldsymbol{M}^A$ for the first layer and (Right) shows result after we use $\boldsymbol{M}^A$ for the first layer and $\boldsymbol{M}^B$ for the second layer.

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	44	45
51	52	53	54	55

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	44	45
51	52	53	54	55

FIGURE 3 – (Left) After using $\boldsymbol{M}^A$ for the first layer. (Right) After using $\boldsymbol{M}^B$ for the second layer.

3. If we use $\boldsymbol{M}^B$ for the first layer and $\boldsymbol{M}^A$ for the second layer.
   Solution: Figure 4 (Left)) shows if we use $\boldsymbol{M}^B$ for the first layer and (Right) shows result after we use $\boldsymbol{M}^A$ for the first layer and $\boldsymbol{M}^A$ for the second layer.

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	44	45
51	52	53	54	55

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	44	45
51	52	53	54	55

FIGURE 4 – (Left) After using $\boldsymbol{M}^B$ for the first layer. (Right) After using $\boldsymbol{M}^A$ for the second layer.

4. If we use $\boldsymbol{M}^B$ for the first layer and $\boldsymbol{M}^B$ for the second layer.
   Solution: After using $\boldsymbol{M}^B$ for the first layer and $\boldsymbol{M}^B$ for the second layer, we get receptive field as Figure 3 (Right).

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	44	45
51	52	53	54	55

11	12	13	14	15
21	22	23	24	25
31	32	33	34	35
41	42	43	44	45
51	52	53	54	55

FIGURE 5 – (Left) After using $\boldsymbol{M}^B$ for the first layer. (Right) After using $\boldsymbol{M}^B$ for the second layer.

Question 3 (6-4)**.** In this question, we study some properties of normalizing flows. Let $X \sim P_X$ and $U \sim P_U$ be, respectively, the distribution of the data and a base distribution (e.g. an isotropic gaussian). We define a normalizing flow as $F : \mathcal{U} \to \mathcal{X}$ parametrized by $\boldsymbol{\theta}$. Starting with P_U and then applying F will induce a new distribution $P_{F(U)}$ (used to match P_X). Since normalizing flows are invertible, we can also consider the distribution $P_{F^{-1}(X)}$.

However, some flows, like planar flows, are not easily invertible in practice. If we use P_U as the base distribution, we can only sample from the flow but not evaluate the likelihood. Alternatively, if we use P_X as the base distribution, we can evaluate the likelihood, but we will not be able to sample.

3.1 Show that $D_{KL}[P_X||P_{F(U)}] = D_{KL}[P_{F^{-1}(X)}||P_U]$. In other words, the forward KL divergence between the data distribution and its approximation can be expressed as the reverse KL divergence between the base distribution and its approximation.
Solution: Using inverse function theorem for probability change of variable and change of variable for below integral:

$$\begin{aligned} D_{KL}[P_X(\boldsymbol{x})||P_{F(U)}(\boldsymbol{x})] &= \int_X p_X(\boldsymbol{x}) \log \left(\frac{p_X(\boldsymbol{x})}{p_{F(U)}(\boldsymbol{x})} \right) d\boldsymbol{x} \\ &= \int_X p_X(\boldsymbol{x}) \log \left(\frac{p_X(\boldsymbol{x})}{p_X(F^{-1}(\boldsymbol{x})|\det J_{F^{-1}}(\boldsymbol{x})|)} \right) d\boldsymbol{x} \\ &= \int_{F^{-1}(X)} p_X(F(\boldsymbol{u}))|\det J_F(\boldsymbol{u})| \log \left(\frac{p_X F(\boldsymbol{u})|\det J_F(\boldsymbol{u})|}{p_U(\boldsymbol{u})} \right) d\boldsymbol{u} \\ &= \int_{F^{-1}(X)} p_{F^{-1}(X)}(\boldsymbol{u}) \log \frac{p_{F^{-1}(X)}(\boldsymbol{u})}{p_U(\boldsymbol{u})} d\boldsymbol{u} \\ &= D_{KL}[P_{F^{-1}(X)}(\boldsymbol{u})||P_U(\boldsymbol{u})] \end{aligned} \tag{3}$$

Hence we prove $D_{KL}[P_X||P_{F(U)}] = D_{KL}[P_{F^{-1}(X)}||P_U]$. If f is invertible, forward divergence between the data distribution and its approximation can be expressed as the reverse KL diver- gence between the base distribution and its approximation i.e $D_{KL}\left[P_X||P_Y\right] = D_{KL}\left[f(P_X)||f(P_Y)\right]$.

3.2 Suppose two scenario: 1) you don't have samples from $p_X(\boldsymbol{x})$, but you can evaluate $p_X(\boldsymbol{x})$, 2) you have samples from $p_X(\boldsymbol{x})$, but you cannot evaluate $p_X(\boldsymbol{x})$. For each scenario, specify if you would use the forward KL divergence $D_{KL}[P_X||P_{F(U)}]$ or the reverse KL divergence $D_{KL}[P_{F(U)}||P_X]$ as the objective to optimize. Justify your answer.
Solution: For scenario 1 when we don't have samples from $p_X(\boldsymbol{x})$, but can evaluate $p_X(\boldsymbol{x})$; reverse KL divergence is used.
Expanding reverse KL divergence:

$$\begin{aligned} \mathcal{L}(\theta) &= D_{KL}\left[P_{F(U)}||P_X\right] \\ &= \mathbb{E}_{p_{F(U)}(\boldsymbol{x})}\left[\log p_{F(U)}(\boldsymbol{x};\theta) - \log p_X(\boldsymbol{x})\right] \\ &= \mathbb{E}_{p_U(\boldsymbol{x})}\left[\log p_U(\boldsymbol{u}) - \log|J_F(\boldsymbol{u})| - \log p_X(F(\boldsymbol{u};\theta))\right] \end{aligned} \tag{4}$$

As we can see from above equation, for reverse KL divergence we only need to evaluate $p_X(x)$ and don't need samples from $p_X(x)$.

For scenario 2 when we have samples from $p_X(x)$, but cannot evaluate $p_X(x)$; forward KL divergence is used.
Expanding forward KL divergence:

$$\begin{aligned}\mathcal{L}(\theta) &= D_{KL}\left[P_X||P_{F(U)}\right] \\ &= \mathbb{E}_{p_X(\boldsymbol{x})}\left[\log p_X(\boldsymbol{x}) - \log p_{F(U)}(\boldsymbol{x};\theta)\right] \\ &= \mathbb{E}_{p_X(\boldsymbol{x})}\left[-\log|J_{F^{-1}}(\boldsymbol{x})| - \log p_U(F^{-1}(\boldsymbol{x};\theta))\right] + \text{const.}\end{aligned} \tag{5}$$

As we can see in above equation that expectation is w.r.t $p_X(x)$ which means we need samples from $p_X(x)$ but no need to evaluate $\log p_X(\boldsymbol{x})$.

Question 4 (4-3-6)**.** In this question, we are concerned with analyzing the training dynamics of GANs. Consider the following value function

$$V(d, g) = dg \tag{6}$$

with $g \in \mathbb{R}$ and $d \in \mathbb{R}$. We will use this simple example to study the training dynamics of GANs.

1. Consider gradient descent/ascent with learning rate α as the optimization procedure to iteratively minimize $V(d, g)$ w.r.t. g and maximize $V(d, g)$ w.r.t. d. We will apply the gradient descent/ascent to update g and d simultaneously. What is the update rule of g and d? Write your answer in the following form

$$[d_{k+1}, g_{k+1}]^\top = A[d_k, g_k]^\top$$

where A is a 2×2 matrix; i.e. specify the value of A.

Solution: To perform gradient descent on $V(d, g)$ w.r.t. g(generator) since we need to minimize $V(d, g)$. Update rule for g:

$$g_{t+1} = g_t - \alpha\frac{\partial V(d, g)}{\partial g} = g_k - \alpha\frac{\partial(dg)}{\partial g} = g_k - \alpha \cdot d_k$$

To perform gradient ascent on $V(d, g)$ w.r.t. d(discriminator) since we need to maximize $V(d, g)$. Update rule for d:

$$d_{t+1} = d_t + \alpha\frac{\partial V(d, g)}{\partial d} = d_k + \alpha\frac{\partial(dg)}{\partial d} = d_k + \alpha \cdot g_k$$

. To update rule of d and g in following form:

$$[d_{k+1}, g_{k+1}]^\top = A[d_k, g_k]^\top$$

$$\begin{bmatrix} d_{k+1} \\ g_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & \alpha \\ -\alpha & 1 \end{bmatrix} \begin{bmatrix} d_k \\ g_k \end{bmatrix}$$

$$[d_{k+1}, g_{k+1}]^\top = \begin{bmatrix} 1 & \alpha \\ -\alpha & 1 \end{bmatrix} [d_k, g_k]^\top$$

$$\text{where } A = \begin{bmatrix} 1 & \alpha \\ -\alpha & 1 \end{bmatrix}$$

2. The optimization procedure you found in 4.1 characterizes a map which has a stationary point [3], what are the coordinates of the stationary points ?
Solution: Stationary points are those where gradients of $\nabla V(d, g)$ is zero.

$$\nabla V(d,g) = \begin{bmatrix} \frac{\partial V(d,g)}{\partial d} \\ \frac{\partial V(d,g)}{\partial g} \end{bmatrix} = \begin{bmatrix} g \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The coordinates of stationary points (g,d)= (0,0)

3. Analyze the eigenvalues of A and predict what will happen to d and g as you update them jointly. In other word, predict the behaviour of d_k and g_k as $k \to \infty$.
Solution: To analyze eigenvalues of A:

$$det\,(A - \lambda \mathrm{I}) = 0$$

$$det\left(\begin{bmatrix} 1 & \alpha \\ -\alpha & 1 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) = 0$$

$$det\left(\begin{bmatrix} 1-\lambda & \alpha \\ -\alpha & 1-\lambda \end{bmatrix}\right) = 0$$

$$(1-\lambda)^2 + \alpha^2 = 0$$

$$(1-\lambda)^2 = -\alpha^2$$

$$(\lambda - 1)^2 = -\alpha^2 \Rightarrow (\lambda - 1) = \pm\sqrt{-\alpha^2}$$

As $(1-\lambda)^2 = (\lambda - 1)^2$ and suppose $i = \sqrt{-1}$

$$\lambda - 1 = \pm i\alpha$$

$$\lambda = 1 \pm i\alpha$$

So let $\lambda_1 = 1 - i\alpha$ and $\lambda_2 = 1 + i\alpha$. Eigenvalues with complex number has rotation so updates will be performed by λ_1 and λ_2 alternatively in parameter space and will never converge to a point as $k \to \infty$.

Question 5 (4-2-8-4-2)**.** In this question, we will see why stop-gradient is critical for non-contrastive SSL methods like SimSiam and BYOL. We will show that removing stop-gradient results in collapsed representations, using the dynamics of SimSiam as our running example.

Consider a two-layer linear SimSiam model with the time-varying weight matrices given by $W(t) \in \mathbb{R}^{n_2 \times n_1}$ and $W_p(t) \in \mathbb{R}^{n_2 \times n_2}$. Note that $W(t)$ corresponds to the weights of the online **and** the target network, while $W_p(t)$ denotes the weights of the predictor. Let $\boldsymbol{x} \in \mathbb{R}^{n_1}$ be an input datapoint and $\boldsymbol{x}_1, \boldsymbol{x}_2$ be the two augmented versions of the input $\boldsymbol{x}$. Also note that in some instances, the dependence on time (t) is omitted for notational simplicity, and the weight matrices are referred to as W and W_p.

3. A stationary point is a point on the surface of the graph (of the function) where all its partial derivatives are zero (equivalently, the gradient is zero). Source: https://en.wikipedia.org/wiki/Stationary_point

Let $\boldsymbol{f}_1 = W\boldsymbol{x}_1$ be the online representation of $\boldsymbol{x}_1$ and $\boldsymbol{f}_2 = W\boldsymbol{x}_2$ be the target representation of $\boldsymbol{x}_2$. The learning dynamics of W and W_p can be obtained by minimizing SimSiam's objective function as shown below:

$$J(W, W_p) = \frac{1}{2}\mathbb{E}_{x_1,x_2}\left[\|W_p\boldsymbol{f}_1 - \text{Stop-Grad}(\boldsymbol{f}_2)\|_2^2\right]. \tag{7}$$

5.1 Show (with proof) that the above objective can be simplified to:

$$J(W, W_p) = \frac{1}{2}\left[\text{tr}(W_p^{\text{T}}W_pF_1) - \text{tr}(W_pF_{12}) - \text{tr}(F_{12}W_p) + \text{tr}(F_2))\right], \tag{8}$$

where $F_1 = \mathbb{E}\left[\boldsymbol{f}_1\boldsymbol{f}_1^{\text{T}}\right] = W(X + X')W^{\text{T}}$, $F_2 = \mathbb{E}\left[\boldsymbol{f}_2\boldsymbol{f}_2^{\text{T}}\right] = W(X + X')W^{\text{T}}$, and $F_{12} = F_{21} = \mathbb{E}\left[\boldsymbol{f}_1\boldsymbol{f}_2^{\text{T}}\right] = WXW^{\text{T}}$. Here, X is the average augmented view of a datapoint $\boldsymbol{x}$ and X' is the covariance matrix of augmented views $\boldsymbol{x}'$ conditioned on $\boldsymbol{x}$ and then averaged over the data $\boldsymbol{x}$, and tr is the Trace operation [4].

Solution:

Consider that :

$$\begin{aligned}(W_p\boldsymbol{f}_1 - \boldsymbol{f}_2)^\top (W_p\boldsymbol{f}_1 - \boldsymbol{f}_2) &= \boldsymbol{f}_1^\top W_p^\top W_p\boldsymbol{f}_1 - \boldsymbol{f}_1^\top W_p^\top \boldsymbol{f}_2 - \boldsymbol{f}_2^\top W_p\boldsymbol{f}_1 + \boldsymbol{f}_2^\top \boldsymbol{f}_2 \\ &= tr\left(W_p^\top W_p\boldsymbol{f}_1\boldsymbol{f}_1^T\right) - tr\left(W_p\boldsymbol{f}_1\boldsymbol{f}_2^\top\right) - tr\left(W_p^\top \boldsymbol{f}_2\boldsymbol{f}_1^\top\right) + tr\left(\boldsymbol{f}_2\boldsymbol{f}_2^T\right)\end{aligned} \tag{9}$$

$$\begin{aligned}J(W, W_p) &= \frac{1}{2}\mathbb{E}_{x_1,x_2}\left[\|W_p\boldsymbol{f}_1 - \text{Stop-Grad}(\boldsymbol{f}_2)\|_2^2\right] \\ J(W, W_p) &= \frac{1}{2}\mathbb{E}_{x_1,x_2}\left[(W_p\boldsymbol{f}_1 - \boldsymbol{f}_2)^\top (W_p\boldsymbol{f}_1 - \boldsymbol{f}_2)\right] \\ &= \frac{1}{2}\mathbb{E}_{x_1,x_2}\left[tr\left(W_p^\top W_p\boldsymbol{f}_1\boldsymbol{f}_1^T\right) - tr\left(W_p\boldsymbol{f}_1\boldsymbol{f}_2^\top\right) - tr\left(W_p^\top \boldsymbol{f}_2\boldsymbol{f}_1^\top\right) + tr\left(\boldsymbol{f}_2\boldsymbol{f}_2^T\right)\right]\end{aligned} \tag{10}$$

Let $F_1 = \mathbb{E}\left[\boldsymbol{f}_1\boldsymbol{f}_1^{\text{T}}\right] = W(X + X')W^{\text{T}}$, $F_2 = \mathbb{E}\left[\boldsymbol{f}_2\boldsymbol{f}_2^{\text{T}}\right] = W(X + X')W^{\text{T}}$, and $F_{12} = F_{21} = \mathbb{E}\left[\boldsymbol{f}_1\boldsymbol{f}_2^{\text{T}}\right] = WXW^{\text{T}}$ and X is the average augmented view of a datapoint $\boldsymbol{x}$ and X' is the covariance matrix of augmented views $\boldsymbol{x}'$ conditioned on $\boldsymbol{x}$ and then averaged over the data $\boldsymbol{x}$, and tr is the Trace.

$$J(W, W_p) = \frac{1}{2}\left[\text{tr}(W_p^{\text{T}}W_pF_1) - \text{tr}(W_pF_{12}) - \text{tr}(F_{12}W_p) + \text{tr}(F_2))\right]$$

Reference: https://arxiv.org/pdf/2102.06810.pdf

5.2 Based on the above expression for $J(W, W_p)$, find the gradient update for W_p (the predictor network), denoting it as $\dot{W}_p$. In other words, obtain an expression for $\dot{W}_p = -\frac{\partial J}{\partial W_p}$ (the derivative of the objective function w.r.t the parameters W_p).

Solution: Taking partial derivative of the objective function w.r.t to parameter W_p:

$$\dot{W}_p = -\frac{\partial J}{\partial W_p}$$

4. https://en.wikipedia.org/wiki/Trace_(linear_algebra).

$$\begin{aligned}\dot{W}_p &= -\frac{1}{2}\frac{\partial\left[\mathrm{tr}(W_p^{\mathrm{T}}W_pF_1) - \mathrm{tr}(W_pF_{12}) - \mathrm{tr}(F_{12}W_p) + \mathrm{tr}(F_2)\right)]}{\partial W_p} \\ &= -\frac{1}{2}\left[W_pF_1 + W_pF_1^\top - F_{12}^\top - F_{12}\right] \\ &= -W_pF_1 + F_{12}^\top\end{aligned} \tag{11}$$

5.3 Consider the case when the Stop-Grad is removed. The gradient of the objective function $J(W, W_p)$ w.r.t the parameters W i.e. $\dot{W}(t) = -\frac{\partial J}{\partial W(t)}$, is given by:

$$\dot{W}(t) = \frac{d}{dt}\mathrm{vec}(W(t)) = -H(t)\mathrm{vec}(W(t)),$$

where $H(t)$ is a time-varying positive semi-definite matrix defined as

$$H(t) = X' \otimes \left(W_p(t)^{\mathrm{T}}W_p(t) + I_{n_2}\right) + X \otimes \left(\tilde{W}_p(t)^{\mathrm{T}}\tilde{W}_p(t)\right).$$

Here, $\otimes$ is the Kronecker product [5] , $\tilde{W}_p(t) = (W_p(t) - I_{n_2})$, and "vec(W)" refers to the *vectorization* of a matrix W [6]. For simplicity, we are not taking weight decay into account here [7].
If the minimal eigenvalue $\lambda_{min}(H(t))$ is bounded away from zero, i.e. $\inf_{t\geq 0}\lambda_{min}(H(t)) \geq \lambda_0 > 0$, then **prove that** $W(t) \to 0$.
Note: In order to prove the above question, the following property must be used:
For a time-varying positive definite matrix $H(t)$ whose minimal eigenvalues are bounded away from 0, the dynamics shown below:

$$\frac{d}{dt}\boldsymbol{w}(t) = -H(t)\boldsymbol{w}(t),$$

satisfies the constraint $\|\boldsymbol{w}(t)\|_2 = e^{-\lambda_0 t}\|\boldsymbol{w}(0)\|_2$, implying that $\boldsymbol{w}(t) \to 0$.
Solution: If we don't have stop gradient then we need to calculate $\frac{\partial J}{\partial F_2}$ for obtaining gradient of the objective function.
Replacing value of $F_1 = W(X + X')W^\top$, $F_2 = W(X + X')W^\top$ and $F_{12} = WXW^\top$ in objective function and calculating gradient:

$$\begin{aligned}\dot{W}(t) &= -\frac{dJ}{d(W(t))} \\ &= -\frac{1/2 * \left[tr(W_p^\top W_pW(X + X')W^\top) - tr(W_pWXW^\top) - tr(WXW^\top) + tr(W(X + X')W^\top)\right]}{d(W(t))} \\ &= \left[-W_p^\top W_pW(X + X') + (W_p^\top + W_p)WX - W(X + X') - W\right] \\ &- (W_p^\top W_p + I)WX' - (W_p^\top W_p - W_p^\top - W_p + I)WX - W \\ &= -(W_p^\top W_p + I)WX' - (W_p - I)^\top(W_p - I)WX - W \\ &= -(W_p^\top W_p + I)WX' - \tilde{W}_p^\top\tilde{W}_pWX - W \left[\because \tilde{W}_p = W_p - I_{n_2}\right]\end{aligned} \tag{12}$$

5. For more information, see https://en.wikipedia.org/wiki/Kronecker_product#Matrix_equations
6. Also known as the "vec trick", it is obtained by stacking all the columns of a matrix A into a single vector.
7. Although omitted here, it must be noted that having weight decay is important. It has also been shown that, in practice, weight decay leads to stable learning.

For $\text{vec}(A \times B) = (B^\top \otimes A)\text{vec}(X)$. Using this we write for $\frac{\mathrm{d}}{\mathrm{d}t}\text{vec}(W)$:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\text{vec}(W(t) &= -\left(X' \otimes (W_p^\top W_p + I_{n_2}) + X \otimes \tilde{W}_p^\top \tilde{W}_p + I_{n_1 n_2}\right)\text{vec}(W) \\ &= -\left(X' \otimes (W_p^\top W_p + I_{n_2}) + X \otimes \tilde{W}_p^\top \tilde{W}_p\right)\text{vec}(W) \qquad (13)\\ \frac{\mathrm{d}}{\mathrm{d}t}\text{vec}(W(t)) &= H(t)\text{vec}(W(t)) \left[\because H(t) = \left(X' \otimes (W_p^\top W_p + I_{n_2}) + X \otimes \tilde{W}_p^\top \tilde{W}_p\right)\right]\end{aligned}$$

Constraint given in question $\|\boldsymbol{w}(t)\|_2 = e^{-\lambda_0 t}\|\boldsymbol{w}(0)\|_2$, implying that $\boldsymbol{w}(t) \to 0$ for dynamics $\frac{d}{dt}\boldsymbol{w}(t) = -H(t)\boldsymbol{w}(t)$. Now using this property for $\frac{d}{dt}W(t) = -H(t)W(t)$ where $\inf_{t\geq 0} \lambda_{min}(H(t)) \geq \lambda_0 > 0$ we have:

$$\|\text{vec}\left(W(t)\right)\|_2^2 \leq e^{-\lambda_0 t}\left\|\text{vec}(W(0))\right\|_2 \to 0$$

which proves $W(t) \to 0$ and W has no possibility of learning any useful features.

5.4 Consider the case when both the Stop-Grad **and** the predictor are removed. Show that the representations collapse i.e. $W(t) \to 0$. You may assume that X' is a positive definite matrix.
Solution: When both the Stop - Grad and predictor are removed i.e $W_p = I$, $W(t)$ always tends to zero. Previous question already proved without Stop-gradient it the case. But if there is Stop gradient:

$$\begin{aligned}\dot{W} &= -(W_p^\top W_p + I)WX' - \tilde{W}_p^\top \tilde{W}_p WX - W \\ &= -WX' - W \,[\because W_p = I] \qquad (14)\\ &= -W\left(X' + I\right)\end{aligned}$$

X' is positive definite matrix and using property from previous question:

$$\|\text{vec}\left(W(t)\right)\|_2^2 \leq e^{-\lambda_0 t}\left\|\text{vec}(W(0))\right\|_2 \to 0$$

Hence $W(t) \to 0$

5.5 Speculate (in 1-2 sentences) as to why the stop-gradient and the predictor are necessary for avoiding representational collapse.
Solution: As seen in previous two question, we have proven analytically that removing either stop-gradient and the predictor leads to representational collapse. Without stopgradient, the optimizer quickly finds a degenerated solution and reaches the minimum possible loss of -1. Recent research says that its due to asymmetrical structure of SimSiam where the role of stop gradient is to only allow the path with predictor to be optimized with the encoder output as the target, not vice versa.

Reference-
Chen, Xinlei, and Kaiming He. "Exploring simple siamese representation learning." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.
-Zhang, Chaoning, et al. "How Does SimSiam Avoid Collapse Without Negative Samples? A Unified Understanding with Self-supervised Contrastive Learning." arXiv preprint arXiv:2203.16262 (2022).
-Tian, Yuandong, Xinlei Chen, and Surya Ganguli. "Understanding self-supervised learning dynamics without contrastive pairs." International Conference on Machine Learning. PMLR, 2021.