Due Date: April 15th, 2022, 11 p.m. EST

Instructions

- *For all questions, show your work!*
- *Please use a document preparation system such as LaTeX.*
- *Unless noted that questions are related, assume that notation and definitions for each question are self-contained and independent.*
- *Submit your answers electronically via Gradescope.*
- *TAs for this assignment are* ***Mohammad-Javad Bayazi*** *and* ***Naga Karthik****.*

This assignment covers mathematical and algorithmic techniques in the families of deep generative models and some of the recent self-supervised learning methods. Thus, we will explore Variational autoencoders (VAEs, Question 1), Autoregressive models (Question 2), Normalizing flows (Question 3), Generative adversarial networks (GANs, Question 4), and Self-supervised models (Question 5).

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 $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_K)$ 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 $p(\boldsymbol{z}|\boldsymbol{x})$.

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.

Answer.

$$\begin{aligned}
&\arg\max_\theta \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})] \\
&= \arg\max_\theta \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})\frac{p_\theta(\boldsymbol{z}|\boldsymbol{x})}{p_\theta(\boldsymbol{z}|\boldsymbol{x})}] \quad \text{using } \frac{p_\theta(\boldsymbol{z}|\boldsymbol{x})}{p_\theta(\boldsymbol{z}|\boldsymbol{x})} = 1 \\
&= \arg\max_\theta \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log \frac{p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{p_\theta(\boldsymbol{z}|\boldsymbol{x})}\frac{p_\theta(\boldsymbol{z}|\boldsymbol{x})}{q(\boldsymbol{z}|\boldsymbol{x})}] \quad \text{as } q(\boldsymbol{z}|\boldsymbol{x}) \text{ independant} \\
&= \arg\max_\theta \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log \frac{p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{p_\theta(\boldsymbol{z}|\boldsymbol{x})} + \log\frac{p_\theta(\boldsymbol{z}|\boldsymbol{x})}{q(\boldsymbol{z}|\boldsymbol{x})}] \\
&= \arg\max_\theta \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log \frac{p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{p_\theta(\boldsymbol{z}|\boldsymbol{x})}] + \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log\frac{p_\theta(\boldsymbol{z}|\boldsymbol{x})}{q(\boldsymbol{z}|\boldsymbol{x})}] \\
&= \arg\max_\theta \log p_\theta(\boldsymbol{x}) - D_{\mathrm{KL}}(q(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x}))
\end{aligned}$$

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 ?

Answer.

$$\begin{aligned}
\mathcal{L}(\theta, q_\phi^*; \boldsymbol{x}_i) &= \log(p_\theta(\boldsymbol{x}_i)) - D_{\mathrm{KL}}(q_\phi^*(\boldsymbol{z}|\boldsymbol{x}_i)||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i)) \\
\mathcal{L}(\theta, q_i^*; \boldsymbol{x}_i) &= \log(p_\theta(\boldsymbol{x}_i)) - D_{\mathrm{KL}}(q_i^*(\boldsymbol{z})||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i))
\end{aligned}$$

Therefore:

$$\begin{aligned}
&\log(p_\theta(\boldsymbol{x}_i)) = \log(p_\theta(\boldsymbol{x}_i)) \\
\Leftrightarrow\ &\mathcal{L}(\theta, q_\phi^*; \boldsymbol{x}_i) + D_{\mathrm{KL}}(q_\phi^*(\boldsymbol{z}|\boldsymbol{x}_i)||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i)) = \mathcal{L}(\theta, q_i^*; \boldsymbol{x}_i) + D_{\mathrm{KL}}(q_i^*(\boldsymbol{z})||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i))
\end{aligned}$$

Thus,

$$D_{\mathrm{KL}}(q_\phi^*(\boldsymbol{z}|\boldsymbol{x}_i)||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i)) = D_{\mathrm{KL}}(q_i^*(\boldsymbol{z})||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i)) + (\mathcal{L}(\theta, q_i^*; \boldsymbol{x}_i) - \mathcal{L}(\theta, q_\phi^*; \boldsymbol{x}_i))$$

Note that $(\mathcal{L}(\theta, q_i^*; \boldsymbol{x}_i) - \mathcal{L}(\theta, q_\phi^*; \boldsymbol{x}_i))$ is positive as q_i^* is the maximiser of the corresponding ELBO. Thus,

$$\boxed{D_{\mathrm{KL}}(q_\phi^*(\boldsymbol{z}|\boldsymbol{x}_i)||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i)) \geq D_{\mathrm{KL}}(q_i^*(\boldsymbol{z})||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i))}$$

■.

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)
Answer. A biais exist there when we try to estimate the marginal log-likelihood using the ELBO. This is the result of the KL Divergence as in the ELBO, the only term depending on θ is the expectation on marginal log-likelihood. Therefore the relation for the biais is the same as in question 1.2.

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

(b) from the computational point of view (efficiency)
Answer. $D_{\mathrm{KL}}(q^*_\phi(\boldsymbol{z}|\boldsymbol{x}_i)||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i))$ is more efficient computationnaly as we only need to compute the posterior $q^*_\phi(\boldsymbol{z}|\boldsymbol{x}_i)$ for q^*_ϕ. Using q^*_i we would have to compute a different posterior for each $\boldsymbol{x}_i$ which is expensive.

(c) in terms of memory (storage of parameters)
Answer. $D_{\mathrm{KL}}(q^*_\phi(\boldsymbol{z}|\boldsymbol{x}_i)||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i))$ is also more memory efficient as we only need to store the parameter set of q^*_ϕ where for $D_{\mathrm{KL}}(q^*_i(\boldsymbol{z})||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i))$ we have to save a set of parameters q^*_i for all data points $\boldsymbol{x}_i$ (which is really expensive).

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.

Answer.

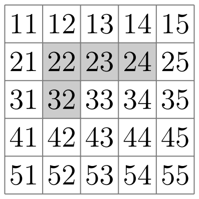

FIGURE 2 – Receptive field of **second layer** when using $\boldsymbol{M}^A$ at index 33 (third row, third column).

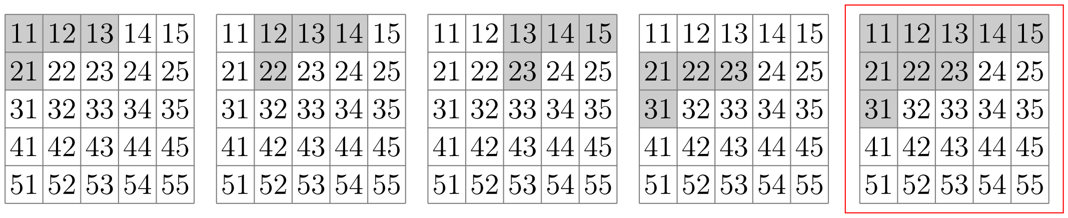

FIGURE 3 – Receptive field of the 22, 23, 24, and 32 pixel of **first layer** when using $\boldsymbol{M}^A$ respectively.Most right image is the receptive field of the output pixel (index) 33 when **first layer** uses $\boldsymbol{M}^B$ and **second layer** uses $\boldsymbol{M}^A$.

2. If we use $\boldsymbol{M}^A$ for the first layer and $\boldsymbol{M}^B$ for the second layer.

Answer.

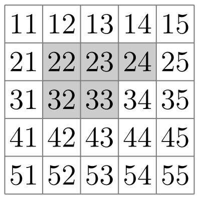

FIGURE 4 – Receptive field of **second layer** when using $\boldsymbol{M}^B$ at index 33 (third row, third column).

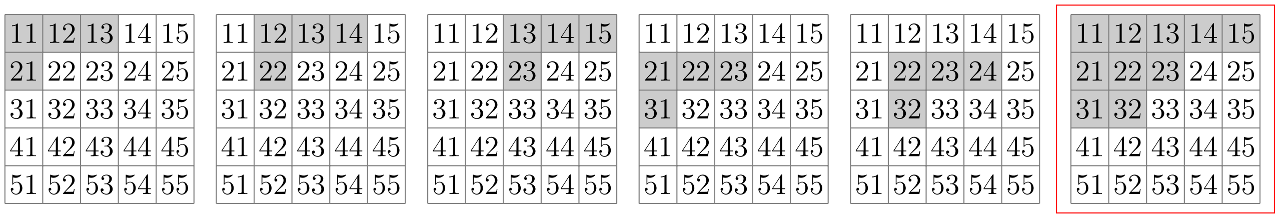

FIGURE 5 – Receptive field of the 22, 23, 24, 32, and 33 pixel of **first layer** when using $\boldsymbol{M}^A$ respectively. Most right image is the receptive field of the output pixel (index) 33 when **first layer** uses $\boldsymbol{M}^A$ and **second layer** uses $\boldsymbol{M}^B$.

3. If we use $\boldsymbol{M}^B$ for the first layer and $\boldsymbol{M}^A$ for the second layer.

Answer.

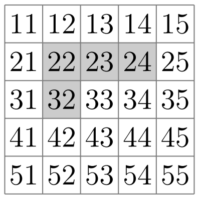

FIGURE 6 – Receptive field of **second layer** when using $\boldsymbol{M}^A$ at index 33 (third row, third column).

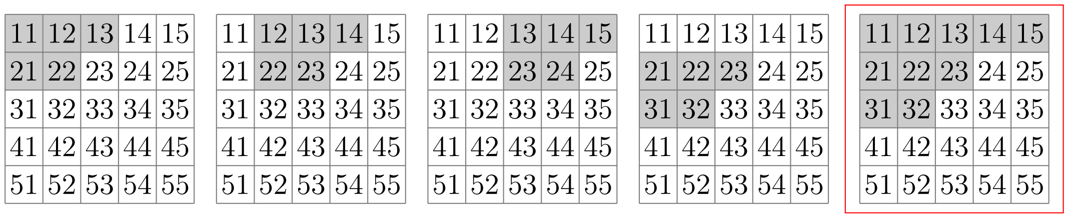

FIGURE 7 – Receptive field of the 22, 23, 24, and 32 pixel of **first layer** when using $\boldsymbol{M}^B$ respectively.Most right image is the receptive field of the output pixel (index) 33 when **first layer** uses $\boldsymbol{M}^B$ and **second layer** uses $\boldsymbol{M}^A$.

4. If we use $\boldsymbol{M}^B$ for the **first layer** and $\boldsymbol{M}^B$ for the second layer.

Answer.

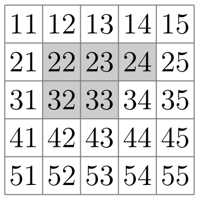

FIGURE 8 – Receptive field of **second layer** when using $\boldsymbol{M}^B$ at index 33 (third row, third column).

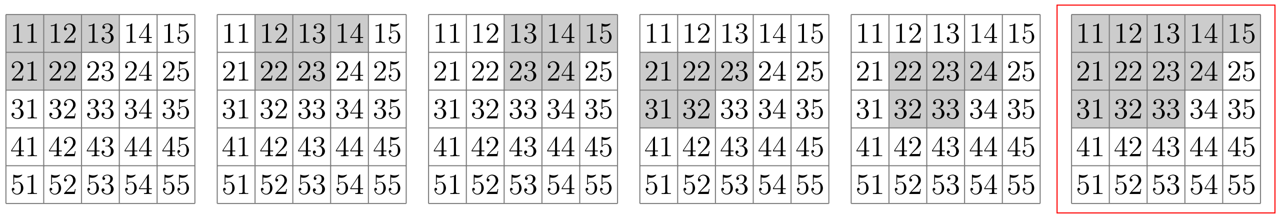

FIGURE 9 – Receptive field of the 22, 23, 24, 32 ,and 33 pixel of **first layer** when using $\boldsymbol{M}^B$ respectively. Most right image is the receptive field of the output pixel (index) 33 when **first layer** uses $\boldsymbol{M}^B$ and **second layer** uses $\boldsymbol{M}^B$.

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.

Answer. First let recall definition of D_{KL}.

$$\begin{aligned}
[P_X||P_{F(U)}] &= \mathbb{E}_{x\sim P_x}[\log(\frac{P_X}{P_{F(U)}})] \\
&= \int_x P_X(x) \log\left(\frac{P_X(x)}{P_{F(U)}(x)}\right) dx \\
&= \int_x P_X(x) \log\left(\frac{P_X(x)}{P_U(F^{-1}(x)).\left|\det \frac{\partial F^{-1}(x)}{\partial x}\right|}\right) dx \\
&= \int_u P_X(F(u)) \log\left(\frac{P_X(F(u)).\left|\det \frac{\partial F(u)}{\partial u}\right|}{P_U(u)}\right).\left|\det \frac{\partial F(u)}{\partial u}\right| du \quad \text{using change of variables}
\end{aligned}$$

where $u \sim P_u$

$$\begin{aligned}
&= \int_u P_{F^{-1}(X)}(u) \log\left(\frac{P_{F^{-1}(X)}(u)}{P_U(u)}\right) du \\
&= D_{\text{KL}}\left[P_{F^{-1}(X)}(u)|P_U(u)\right] \\
&= \boxed{D_{\text{KL}}\left[P_{F^{-1}(X)}|P_U\right]}
\end{aligned}$$

■.

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.

Answer.

Scenario n°1: With reverse KL divergence $D_{\text{KL}}[P_{F(u)}||P_X]$, we are able to evaluate P_X but we don't have samples as the KL divergence equation does not contains a term that require to sample from P_X. That suits well for our case as we doesn't have samples and can evaluate $P_X(x)$.

Scenario n°2: Logically, as the forward KL divergence $D_{\mathrm{KL}}[P_X||P_{F(U)}]$ rely on inverse operation meaning that we need samples to compute the KL divergence but we don't rely on evaluation using P_X. As in that scenario we have samples, we can use the forward KL divergence equation.

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

$$V(d, g) = dg \tag{1}$$

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.

Answer.
First recall that: Critic update:

$$d_{k+1} = d_k + \alpha \frac{\partial}{\partial d} V(d_k, g_k) = d_k + \alpha g_k$$

Generator update:

$$g_{k+1} = g_k - \alpha \frac{\partial}{\partial g} V(d_k, g_k) = g_k - \alpha d_k$$

Therefore to make the update simultaneously we need to define:

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

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

2. The optimization procedure you found in 6.1 characterizes a map which has a stationary point [3], what are the coordinates of the stationary points ?

Answer. Reminding that stationary point is a point where the gradient is equal to 0 for every dimensions, i.e.

$$\frac{\partial V(d,g)}{\partial d} = 0 \text{ and } \frac{\partial V(d,g)}{\partial g} = 0 \implies [0,0]^\top$$

Coordinates of the stationary point are $(0, 0)$.

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

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 $k \to \infty$.

Answer. To find eigenvalues, first remind the equation:

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

Therefore,

$$\begin{aligned} &\det(\boldsymbol{A} - \lambda \boldsymbol{I}) = 0 \\ \Leftrightarrow &\det\left(\begin{bmatrix} 1-\lambda & \alpha \\ -\alpha & 1-\lambda \end{bmatrix}\right) = 0 \\ \Leftrightarrow &(1-\lambda)(1-\lambda) + \alpha^2 = 0 \\ \Leftrightarrow &\lambda^2 - 2\lambda + 1 + \alpha^2 = 0 \end{aligned}$$

Using formula to solve second degree equation: $\Delta = b^2 - 4ac$,

$$\begin{aligned} &\Delta = b^2 - 4ac = (-2)^2 - 4.1.(1+\alpha^2) \\ \Leftrightarrow &\Delta = 4 - 4 - 4\alpha^2 = -4\alpha^2 = 4i^2\alpha^2 \end{aligned}$$

Root:

$$\begin{aligned} \frac{-b \pm \sqrt{\Delta}}{2a} &= \frac{2 \pm \sqrt{4i^2\alpha^2}}{2} \\ &= \frac{2 \pm 2i\alpha}{2} = \boxed{1 \pm i\alpha} \qquad \text{where } i = \sqrt{-1} \end{aligned}$$

Therefore the two eigenvalues are $\lambda_1 = 1 + i\alpha$ and $\lambda_2 = 1 - i\alpha$ where α is the learning rate.

Complex eigenvalues means scaling eigen-vector by a real and a rotation (due to imaginary part). Therefore, as $k \to \infty$, and because of their update rules, g and d will rotate indefinitely around stationary point (0,0) and will never converge.

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.

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:

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

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

$$J(W, W_p) = \frac{1}{2}\left[\operatorname{tr}(W_p^{\mathrm{T}} W_p F_1) - \operatorname{tr}(W_p F_{12}) - \operatorname{tr}(F_{12} W_p) + \operatorname{tr}(F_2))\right], \tag{3}$$

where $F_1 = F_2 = \mathbb{E}\left[\boldsymbol{f}_1 \boldsymbol{f}_1^{\mathrm{T}}\right] = W(X + X')W^{\mathrm{T}}$ and $F_{12} = F_{21} = \mathbb{E}\left[\boldsymbol{f}_1 \boldsymbol{f}_2^{\mathrm{T}}\right] = WXW^{\mathrm{T}}$. 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].

Answer.

$$\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] \\
&= \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[((W_p\boldsymbol{f}_1)^\top - \boldsymbol{f}_2^\top)(W_p\boldsymbol{f}_1 - \boldsymbol{f}_2)\right] \\
&= \frac{1}{2}\mathbb{E}_{x_1,x_2}\left[\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\right] \\
&= \frac{1}{2}\mathbb{E}_{x_1,x_2}\left[\operatorname{tr}(\boldsymbol{f}_1^\top W_p^\top W_p\boldsymbol{f}_1) - \operatorname{tr}(\boldsymbol{f}_1^\top W_p^\top \boldsymbol{f}_2) - \operatorname{tr}(\boldsymbol{f}_2^\top W_p\boldsymbol{f}_1) + \operatorname{tr}(\boldsymbol{f}_2^\top\boldsymbol{f}_2)\right] \\
&= \frac{1}{2}\mathbb{E}_{x_1,x_2}\left[\operatorname{tr}(W_p^\top W_p\boldsymbol{f}_1\boldsymbol{f}_1^\top) - \operatorname{tr}(W_p^\top \boldsymbol{f}_2\boldsymbol{f}_1^\top) - \operatorname{tr}(\boldsymbol{f}_1\boldsymbol{f}_2^\top W_p) + \operatorname{tr}(\boldsymbol{f}_2\boldsymbol{f}_2^\top)\right] \\
&= \frac{1}{2}\left[\mathbb{E}[\operatorname{tr}(W_p^\top W_p\boldsymbol{f}_1\boldsymbol{f}_1^\top)] - \mathbb{E}[\operatorname{tr}(W_p^\top \boldsymbol{f}_2\boldsymbol{f}_1^\top)] - \mathbb{E}[\operatorname{tr}(\boldsymbol{f}_1\boldsymbol{f}_2^\top W_p)] + \mathbb{E}[\operatorname{tr}(\boldsymbol{f}_2\boldsymbol{f}_2^\top)]\right] \\
&= \frac{1}{2}\left[\operatorname{tr}(\mathbb{E}[W_p^\top W_p\boldsymbol{f}_1\boldsymbol{f}_1^\top]) - \operatorname{tr}(\mathbb{E}[W_p^\top \boldsymbol{f}_2\boldsymbol{f}_1^\top]) - \operatorname{tr}(\mathbb{E}[\boldsymbol{f}_1\boldsymbol{f}_2^\top W_p]) + \operatorname{tr}(\mathbb{E}[\boldsymbol{f}_2\boldsymbol{f}_2^\top])\right] \\
&= \frac{1}{2}\left[\operatorname{tr}(W_p^\top W_p\mathbb{E}[\boldsymbol{f}_1\boldsymbol{f}_1^\top]) - \operatorname{tr}(W_p^\top \mathbb{E}[\boldsymbol{f}_2\boldsymbol{f}_1^\top]) - \operatorname{tr}(\mathbb{E}[\boldsymbol{f}_1\boldsymbol{f}_2^\top]W_p) + \operatorname{tr}(\mathbb{E}[\boldsymbol{f}_2\boldsymbol{f}_2^\top])\right] \\
&= \frac{1}{2}\left[\operatorname{tr}(W_p^\top W_p F_1) - \operatorname{tr}(W_p^\top F_{21}) - \operatorname{tr}(F_{12}W_p) + \operatorname{tr}(F_2)\right] \\
&= \frac{1}{2}\left[\operatorname{tr}(W_p^\top W_p F_1) - \operatorname{tr}(W_p F_{12}) - \operatorname{tr}(F_{12}W_p) + \operatorname{tr}(F_2)\right]
\end{aligned}$$

■.

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).

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

Answer.

$$\begin{aligned}
\dot{W}_p &= -\frac{\partial J}{\partial W_p} \\
&= -\frac{\partial}{\partial W_p}\frac{1}{2}[\mathrm{tr}(W_p^\top W_p F_1) - \mathrm{tr}(W_p F_{12}) - \mathrm{tr}(W_p F_{12}) - \mathrm{tr}(W_p F_{12}) - \mathrm{tr}(F_2)] \\
&= -\frac{1}{2}[\frac{\partial}{\partial W_p}\mathrm{tr}(W_p^\top W_p F_1) - \frac{\partial}{\partial W_p}\mathrm{tr}(W_p F_{12}) - \frac{\partial}{\partial W_p}\mathrm{tr}(W_p F_{12}) - \frac{\partial}{\partial W_p}\mathrm{tr}(F_2)] \\
&= -\frac{1}{2}[W_p F_1^\top - 2F_{12}^\top] \\
&= -W_p F_1^\top + F_{12}^\top = -W_p F_1 + F_{12}^\top \quad \text{as } F_1 \text{ is a symetric matrix}
\end{aligned}$$

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** $W(t) \to 0$.
<u>**Note:**</u> 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 $\boldsymbol{w}(t) \to 0$.

Answer. First, let recall the definition of the Kronecker Product:

$$\mathrm{vec}(AXB) = \left(B^\top \otimes A\right)\mathrm{vec}(X) \tag{4}$$

$$\begin{aligned}
\dot{W}(t) = -\frac{\partial J}{\partial W(t)} &= -W_p(t)^\top W_p(t) W(t)\left(X + X'\right) + \left(W_p(t)^\top + W_p(t)\right) W(t) X - W(t)\left(X + X'\right) \\
&= -\left(W_p(t)^\top W_p(t) + I_{n_2}\right) W(t) X' - \left(W_p(t)^\top W_p(t) - W_p(t)^\top - W_p(t) + I_{n_2}\right) W(t) X \\
&= -\left(W_p(t)^\top W_p(t) + I_{n_2}\right) W(t) X' - \left(W_p(t) - I_{n_2}\right)^\top \left(W_p(t) - I_{n_2}\right) W(t) X \\
&= -\left(W_p(t)^\top W_p(t) + I_{n_2}\right) W(t) X' - \tilde{W}_p(t)^\top \tilde{W}_p(t) W(t) X
\end{aligned}$$

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.

Then, using the definition of the **Kronecker Product** (4):

$$\frac{d}{dt}\operatorname{vec}(W(t)) = -\left[X' \otimes \left(W_p(t)^\top W_p(t) + I_{n_2}\right) + X \otimes (\tilde{W}_p(t)^\top \tilde{W}_p(t))\right]\operatorname{vec}(W(t))$$

Thus, using the property given in the **Note**, we can prove that $\boldsymbol{w}(t) \to 0$ and therefore collaspe.

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.

Answer.
Now that we don't have stop-grad nor predictor, we can replace $W_p = I_{n_2}$

Therefore,

$$\begin{aligned}
\dot{W}(t) &= -\frac{\partial J}{\partial W(t)} \\
&= -H(t)\operatorname{vec}(W(t)) \\
&= -\left[X' \otimes (I_{n_2} + I_{n_2}) + X \otimes ((I_{n_2} - I_{n_2})^\top (I_{n_2} - I_{n_2}))\right]\operatorname{vec}(W(t)) \\
&= -\left[X' \otimes (I_{n_2} + I_{n_2})\right]\operatorname{vec}(W(t)) \\
&= -\left[X' \otimes 2I_{n_2}\right]\operatorname{vec}(W(t))
\end{aligned}$$

Thus, based on the proof of q5.2 and the fact that X' is a positive definite matrix, we can show that $W(t) \to 0$.

5.5 Speculate (in 1-2 sentences) as to why the stop-gradient and the predictor are necessary for avoiding representational collapse.

Answer.
After doing the exerice, a valid mathematical interpreation is:
In a mathematical point of view, the combination of stop-gradient and predictor avoid representational collapse as it **avoids** the dynamic to converge towards 0 i.e. **not** $\boldsymbol{w}(t) \to 0$.

Note : Personal Interpretation (intuition ?)
(Not sure if it is valid to intepрrate it like that, and this is more a comment than my answer)
In a more "deep learning" point of view, I would interpret this as adding the predictor leads to learn different representation in the neural network using it as it has more capacity when having the predictor and can therefore allocate differently how to representation are learn. Furthermore, having stop-gradient make sure that the two backbone neural networkds have delayed representation in time (using EMA for instance).