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

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 $p(\boldsymbol{z}|\boldsymbol{x})$.

1.2 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?

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

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.

(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)

(b) from the computational point of view (efficiency)

(c) in terms of memory (storage of parameters)

Answer 1.

1.1 Using the fact that $q(\boldsymbol{z}|\boldsymbol{x})$ does not depend on θ and the identity $1 = p_\theta(\boldsymbol{z}|\boldsymbol{x})/p_\theta(\boldsymbol{z}|\boldsymbol{x})$, we have

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

That it, maximizer of the expected complete data log likelihood aims at maximizing the marginal likelihood and reducing the KL divergence between $q(\boldsymbol{z}|\boldsymbol{x})$ and $p_\theta(\boldsymbol{z}|\boldsymbol{x})$; the latter gap causes the bias.

1.2 Denote by $\mathcal{L}[q(\boldsymbol{z})]$ the lower bound corresponding to the variational distribution $q(\boldsymbol{z})$. Due to the finiteness of the recognition model, the distributions q_ϕ can represent is a subset of $\mathcal{Q}$. Given q_i^* the maximizer of the ELBO (argmax over the family $\mathcal{Q}$), the inference gap of q_{ϕ^*} can be decomposed as the sum of two positive terms:

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

Using the equality $\log p_\theta(\boldsymbol{x}) - \mathcal{L}[q(\boldsymbol{z}|\boldsymbol{x})] = D_{\mathrm{KL}}(q(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x}))$, we have

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

The last term is non-negative since q_i^* is the maximizer among $\mathcal{Q}$, so the KL on the LHS is no smaller than the KL on the RHS.

1.3 (a) Since the only term in the ELBO that depends on θ is the expected complete data log likelihood, there is a bias due to the KL divergence. As demonstrated by the last question, the recognition model cannot do better than the optimal q_i^*.

(b) Computationally, the point of using a recognition model is to amortize the cost of inference, so that one does not need to update q_i for each data point $\boldsymbol{x}_i$ until the approximation is good enough ; instead one can simply use the output of the recognition model as a reasonable approximation, since the encoder, which is normally jointly trained with the decoder, is constantly updated.

(c) In terms of number of parameters, amortization allows for $\mathcal{O}(1)$ storage (considering a fixed size recognition model) instead of learning a new $q_i(\boldsymbol{z})$ for each $\boldsymbol{x}_i$, which is $\mathcal{O}(n)$.

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,

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

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.

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

Your answer should look like Figure 1 (Right).

Answer 2. See Figure 2

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

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 – Receptive field under different masking schemes.

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.

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 3. 3.1 We use the probability change of variable formula and perform a change of variable for the integral (and use the inverse function theorem):

$$\begin{aligned}
D_{KL}[p_X(\boldsymbol{x})||p_{F(U)}(\boldsymbol{x})] &= \int_{\mathcal{X}} p_X(\boldsymbol{x}) \log \frac{p_X(\boldsymbol{x})}{p_{F(U)}(\boldsymbol{x})} d\boldsymbol{x} \\
&= \int_{\mathcal{X}} p_X(\boldsymbol{x}) \log \frac{p_X(\boldsymbol{x})}{p_U(F^{-1}(\boldsymbol{x}))|\det J_{F^{-1}}(\boldsymbol{x})|} d\boldsymbol{x} \\
&= \int_{F^{-1}(\mathcal{X})} p_X(F(\boldsymbol{u}))|\det J_F(\boldsymbol{u})| \log \frac{p_X(F(\boldsymbol{u}))|\det J_F(\boldsymbol{u})|}{p_U(\boldsymbol{u})} d\boldsymbol{u} \\
&= \int_{F^{-1}(\mathcal{X})} p_{F^{-1}(X)}(\boldsymbol{u}) \log \frac{p_{F^{-1}(X)}(\boldsymbol{u})}{p_U(\boldsymbol{u})} d\boldsymbol{u} \\
&= D_{KL}[p_{F^{-1}(X)}(\boldsymbol{u})||p_U(\boldsymbol{u})]
\end{aligned}$$

Note that the equality also holds for the reverse: $D_{KL}[P_{F(U)}||P_X] = D_{KL}[P_U||P_{F^{-1}(X)}]$. In general, it is also true that f-divergences (the KL divergence is a particular case) are invariable to invertible transformation, meaning that $D_{KL}[P_X||P_Y] = D_{KL}[f(P_X)||f(P_Y)]$ if f is invertible.

3.2 The forward KL divergence is used when we have samples from the distribution P_X, but we cannot evaluate p_X. On the other hand, the reverse KL divergence is used when we can evaluate p_X. It becomes apparent if we expand the two terms. For the forward KL divergence, we have:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}) &= D_{KL}[p_X(\boldsymbol{x})||p_{F(U)}(\boldsymbol{x})] \\
&= \mathbb{E}_{p_X(\boldsymbol{x})}[-\log p_{F(U)}(\boldsymbol{x})] + cst. \\
&= -\mathbb{E}_{p_X(\boldsymbol{x})}[\log p_U(F^{-1}(\boldsymbol{x})) + \log|J_{F^{-1}}(\boldsymbol{x})|] + cst.
\end{aligned}$$

that only require $\boldsymbol{x}$. The second term is denoted as a constant since it does not depend on $\boldsymbol{\theta}$. For the reverse KL divergence, we have:

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

that only require to evaluate p_X.

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

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

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.
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?
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 4. 1. $A = \begin{bmatrix} 1 & \alpha \\ -\alpha & 1 \end{bmatrix}$

2. Stationary point: $(0, 0)$
3. The eigenvalues are $\lambda = 1 \pm \sqrt{-\alpha^2}$. It tells us that the trajectory optimization procedure follows a rotation orbit around the fixed point while diverging.

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:
$$J(W, W_p) = \frac{1}{2}\left[\text{tr}(W_p^{\mathrm{T}}W_pF_1) - \text{tr}(W_pF_{12}) - \text{tr}(F_{12}W_p) + \text{tr}(F_2))\right], \tag{3}$$
where $F_1 = \mathbb{E}\left[\boldsymbol{f}_1\boldsymbol{f}_1^{\mathrm{T}}\right] = W(X + X')W^{\mathrm{T}}$, $F_2 = \mathbb{E}\left[\boldsymbol{f}_2\boldsymbol{f}_2^{\mathrm{T}}\right] = W(X + X')W^{\mathrm{T}}$ and $F_{12} = F_{21} = \mathbb{E}\left[\boldsymbol{f}_1\boldsymbol{f}_2^{\mathrm{T}}\right] = WXW^{\mathrm{T}}$. Here, $X = \mathbb{E}\left[\bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^{\mathrm{T}}\right]$, where $\bar{\boldsymbol{x}}$ 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].

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

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

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).

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:

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

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 $\boldsymbol{w}(t) \to 0$.

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.

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

Answer 5. 5.1 The L_2- norm squared error inside the expectation can be written as:

$$\begin{aligned}
\|W_p\boldsymbol{f}_1 - \boldsymbol{f}_2\|_2^2 &= (W_p\boldsymbol{f}_1 - \boldsymbol{f}_2)^{\mathrm{T}} (W_p\boldsymbol{f}_1 - \boldsymbol{f}_2) \\
&= \left(\boldsymbol{f}_1^{\mathrm{T}} W_p^{\mathrm{T}} - \boldsymbol{f}_2^{\mathrm{T}}\right) (W_p\boldsymbol{f}_1 - \boldsymbol{f}_2) \\
&= \boldsymbol{f}_1^{\mathrm{T}} W_p^{\mathrm{T}} W_p \boldsymbol{f}_1 - \boldsymbol{f}_1^{\mathrm{T}} W_p^{\mathrm{T}} \boldsymbol{f}_2 - \boldsymbol{f}_2^{\mathrm{T}} W_p \boldsymbol{f}_1 + \boldsymbol{f}_2^{\mathrm{T}} \boldsymbol{f}_2
\end{aligned}$$

Using the property of Trace that for any vector $\boldsymbol{a}$, $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{a} = \text{tr}(\boldsymbol{a}\boldsymbol{a}^{\mathrm{T}})$ and for any matrix $\boldsymbol{A}$, $\text{tr}(\boldsymbol{A}) = \text{tr}(\boldsymbol{A}^{\mathrm{T}})$, we have,

$$\begin{aligned}
&= \text{tr}(W_p^{\mathrm{T}} W_p \boldsymbol{f}_1 \boldsymbol{f}_1^{\mathrm{T}}) - \text{tr}(W_p^{\mathrm{T}} \boldsymbol{f}_2 \boldsymbol{f}_1^{\mathrm{T}}) - \text{tr}(W_p \boldsymbol{f}_1 \boldsymbol{f}_2^{\mathrm{T}}) + \text{tr}(\boldsymbol{f}_2 \boldsymbol{f}_2^{\mathrm{T}}) \\
&= \text{tr}(W_p^{\mathrm{T}} W_p \boldsymbol{f}_1 \boldsymbol{f}_1^{\mathrm{T}}) - \text{tr}(\boldsymbol{f}_1 \boldsymbol{f}_2^{\mathrm{T}} W_p) - \text{tr}(W_p \boldsymbol{f}_1 \boldsymbol{f}_2^{\mathrm{T}}) + \text{tr}(\boldsymbol{f}_2 \boldsymbol{f}_2^{\mathrm{T}})
\end{aligned}$$

Given the definitions of F_1, F_2, and F_{12} in the question,

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

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.

5.2 First, note that X' is the covariance matrix and X is the auto-correlation matrix, hence, they are symmetric by definition. As a consequence, $F_1 = \mathbb{E}\left[\boldsymbol{f}_1 \boldsymbol{f}_1^{\mathrm{T}}\right]$ is also symmetric, implying that $F_1 = F_1^{\mathrm{T}}$.

Now, using equations 113 and 100 from the Matrix Cookbook we expand the partial derivatives, giving the gradient update for W_p as follows:

$$\begin{aligned}\dot{W}_p = -\frac{\partial J}{\partial W_p} &= -\frac{1}{2}\left[W_p F_1^{\mathrm{T}} + W_p F_1 - 2F_{12}^{\mathrm{T}}\right] \\ &= -\frac{1}{2}\left[2W_p F_1 - 2F_{12}^{\mathrm{T}}\right] \\ &= -W_p F_1 + F_{12}^{\mathrm{T}}\end{aligned}$$

5.3 Considering the simplified form of the objective function and plugging the values of F_1, F_2, and F_{12}, we obtain:

$$J(W, W_p) = \frac{1}{2}\left[\mathrm{tr}(W_p^{\mathrm{T}} W_p W(X + X')W^{\mathrm{T}}) - \mathrm{tr}(W_p W X W^{\mathrm{T}}) - \mathrm{tr}(W X W^{\mathrm{T}} W p) + \mathrm{tr}(W(X + X')W^{\mathrm{T}}))\right]$$

Now, with the stop gradient removed, consider the partial derivatives for each of the four terms above separately (for simplicity). We denote $W_p^{\mathrm{T}} W_p$ as B and $(X + X')$ as C in what follows.

1st Term: We have:

$$\begin{aligned}\frac{\partial}{\partial W}\mathrm{tr}(BWCW^{\mathrm{T}}) &= \frac{\partial}{\partial W}\mathrm{tr}(W^{\mathrm{T}} BWC) \\ &= (BWC) + (B^{\mathrm{T}} WC^{\mathrm{T}}) \quad \text{(Eq. 117 from Matrix Cookbook)}\end{aligned}$$

Note that $B = B^{\mathrm{T}}$ and $C = C^{\mathrm{T}}$ as X and X' are themselves symmetric by definition. Therefore,

$$= 2(BWC) = \underline{2W_p^{\mathrm{T}} W_p W(X + X')}$$

2nd Term: We have $\mathrm{tr}(W_p W X W^{\mathrm{T}}) = \mathrm{tr}(W^{\mathrm{T}} W_p W X)$. Then,:

$$\frac{\partial}{\partial W}\mathrm{tr}(W^{\mathrm{T}} W_p W X) = (W_p W X) + (W_p^{\mathrm{T}} W X^{\mathrm{T}}) \quad \text{(Eq. 117 from Matrix Cookbook)}$$

3rd Term: We have $\mathrm{tr}(W X W^{\mathrm{T}} W_p) = \mathrm{tr}(W_P W X W^{\mathrm{T}}) = \mathrm{tr}(W^{\mathrm{T}} W_p W X)$. Then,

$$\frac{\partial}{\partial W}\mathrm{tr}(W^{\mathrm{T}} W_p W X) = (W_p W X) + (W_p^{\mathrm{T}} W X^{\mathrm{T}}) \quad \text{(Eq. 117 from Matrix Cookbook)}$$

It must be noted that $X = X^{\mathrm{T}}$. Then, adding the simplified forms of the 2nd and the 3rd terms, they can further simplified as:

$$2(W_p W X) + 2(W_p^{\mathrm{T}} W X) = \underline{2(W_p + W_p^{\mathrm{T}})WX}$$

4th Term: We have $\mathrm{tr}(W(X + X')W^{\mathrm{T}}) = \mathrm{tr}(W^{\mathrm{T}} W(X + X')) = \mathrm{tr}(W^{\mathrm{T}} W C)$. Then,

$$\frac{\partial}{\partial W}\mathrm{tr}(W^{\mathrm{T}} W C) = WC^{\mathrm{T}} + WC = 2WC = \underline{2W(X + X')} \quad (\text{as } C = C^{\mathrm{T}})$$

Finally, adding all the individual terms from above, we can obtain the gradient of W from the partial derivative $\frac{\partial J}{\partial W}$ as follows:

$$\begin{aligned}\dot{W} = -\frac{\partial J}{\partial W} &= -\left[W_p^{\mathrm{T}} W_p W(X + X') - (W_p^{\mathrm{T}} + W_p)WX + W(X + X')\right] \\ &= -\left[W_p^{\mathrm{T}} W_p WX + W_p^{\mathrm{T}} W_p WX' - W_p^{\mathrm{T}} WX - W_p WX + WX + WX')\right]\end{aligned}$$

Rearranging all the like terms together, we have,

$$\begin{aligned}&= -\left[(W_p^{\mathrm{T}} W_p + I)WX' + (W_p^{\mathrm{T}} W_p - W_p^{\mathrm{T}} - W_p + I)WX\right] \\ &= -\left[(W_p^{\mathrm{T}} W_p + I)WX' + (W_p - I)^{\mathrm{T}}(W_p - I)WX\right] \\ &= -\left[(W_p^{\mathrm{T}} W_p + I)WX' + \tilde{W_p}^{\mathrm{T}} \tilde{W_p} WX\right]\end{aligned}$$

From the properties of the Kronecker Product, we have that $\mathrm{vec}(AXB) = (B^{\mathrm{T}} \otimes A)\mathrm{vec}(X)$. Therefore, we can rewerite the above expression as:

$$\begin{aligned}&= -\left[X' \otimes (W_p^{\mathrm{T}} W_p + I) + X \otimes (\tilde{W_p}^{\mathrm{T}} \tilde{W_p})\right] \mathrm{vec}(W) \\ &= -H(t)\mathrm{vec}(W)\end{aligned}$$

Therefore, we now have an equation of the form:

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

and since it is given that $\inf_{t \geq 0} \lambda_{min}(H(t)) \geq \lambda_0 > 0$, we have the following constraint which is satisfied:

$$\|\mathrm{vec}(W(t))\|_2 = e^{-\lambda_0 t} \|\mathrm{vec}(W(0))\|_2,$$

thereby implying that $W(t) \to 0$, and hence showing that the representations collapse when the Stop-Grad is removed.

5.4 Since the predictor is also removed, we have $W_p = I$ and we substitute this in equation 3, now given by:

$$\begin{aligned}J(W) &= \frac{1}{2}\left[\mathrm{tr}(F_1) - \mathrm{tr}(F_{12}) - \mathrm{tr}(F_{12}) + \mathrm{tr}(F_2))\right] \\ &= \mathrm{tr}(F_1) - \mathrm{tr}(F_{12}) \\ &= \mathrm{tr}(W(X + X')W^{\mathrm{T}}) - \mathrm{tr}(WXW^{\mathrm{T}})\end{aligned}$$

Similar to the method shown above, the gradient of W is given by:

$$\begin{aligned}\dot{W} &= -\left[2WX + 2WX' - 2WX\right] \\ &= -2WX'\end{aligned}$$

Since X' is a positive definite matrix, using the similar arguments as shown in the previous sub-question, we have that $W(t) \to 0$.

5.5 Maybe SimSiam is a form of the Expectation-Maximization algorithm where there are 2 sets of variables being optimized ; the 2nd set of variables in this case arise from the stop-gradient operation.

(open-ended question, solution not limited to the one above)