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

Answer 1.1

We have $q(\boldsymbol{z}|\boldsymbol{x})$ does not depend on θ, then :

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

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.

Using Bayes theorem, we have : $\frac{p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{p_\theta(\boldsymbol{z}|\boldsymbol{x})} = p_\theta(\boldsymbol{x})$, replacing this in the equation :

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

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 1.2

We have :

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

From the lectures notes, we can write :

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

Replacing in the equality, we get :

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

Since q_i^* is the maximizer amongst Q, the term $(\mathcal{L}[q_i^*(\boldsymbol{z})] - \mathcal{L}[q_{\phi^*}(\boldsymbol{z}|\boldsymbol{x}_i)])$ is positive.
Thus :

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

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)

The bias due to the KL divergence is defined as : $\text{bias}(\theta) = -D_{\mathrm{KL}}(q(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x}))$, so from the question 1.2 :

$$D_{\mathrm{KL}}(q_i^*(\boldsymbol{z})||p_\theta(\boldsymbol{z}|\boldsymbol{x})) \leq D_{\mathrm{KL}}(q_{\phi^*}(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})) \implies \text{bias}_{x_i,\phi^*}(\theta) \leq \text{bias}_{x_i,q_i^*}(\theta) \leq 0$$

From that the bias is closer to zero when we estimate the optimal q_i^* compared to when we calculate ϕ^* for recognition model. Thus the optimal q_i^* can allow us to have a more accurate estimation of the marginal likelihood vie the ELBO.

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

Answer (b)

Calculating q_i^* for each data point $\boldsymbol{x}_i$ might be computationally heavy, while the recognition model does not need to update q_i for each data point $\boldsymbol{x}_i$ until getting a good approximation. Thus approximating q_{ϕ^*} is more efficient.

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

Answer (c)

Approximating q_i^* needs more memory storage since we store the parameters for each training data point $\boldsymbol{x}_i$, which means n times more expensive than the amortization.

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.

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

Since the padding size is equal to 1 and without kernel flipping, we can see from figure 3 the receptive field in feature map obtained after the first layer. We only consider pixels that actually

k_{11}	k_{12}	k_{13}
k_{21}	0	0
0	0	0

k_{11}	k_{12}	k_{13}
k_{21}	k_{22}	0
0	0	0

FIGURE 2 – (Left) Kernel obtained when apply the mask of type A. (Right) Kernel obtained when apply the mask of type B.

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

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 – Receptive field in feature map 1 (Left) if we use $\boldsymbol{M}^A$. (Right) if we use $\boldsymbol{M}^B$.

participate in the calculation of pixel of index 33 in the output. The kernel masked by type A has 5 inactive pixels and 4 active pixels, while the kernel masked by type B has 4 inactive pixels and 5 active pixels. If we use mask of type A in the second layer, the value of the pixel of index 33 from the output would be calculated as : $k_{11} * fmap_{22} + k_{12} * fmap_{23} + k_{13} * fmap_{24} + k_{21} * fmap_{32}$.

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

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 – Receptive field 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 5 :

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 – Receptive field 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 6 :

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 6 – Receptive field 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 7 :

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 7 – Receptive field if we use $\boldsymbol{M}^B$ for the first layer and $\boldsymbol{M}^B$ 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.

Answer 3.1

We have :

$$D_{KL}[p_X(\boldsymbol{x})||p_{F(U)}(\boldsymbol{x})] = \int_{\mathcal{X}} p_X(\boldsymbol{x}) \log \frac{p_X(\boldsymbol{x})}{p_{F(U)}(\boldsymbol{x})} d\boldsymbol{x}$$

Using the change of variable formula, we have : $p_{F(U)}(\boldsymbol{x}) = p_U(F^{-1}(\boldsymbol{x}))|\det J_{F^{-1}}(\boldsymbol{x})|$ Replacing this in the equation :

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

Let's pose $\boldsymbol{u} = F^{-1}(\boldsymbol{x})$ and using the change of variable formula, we have :

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

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 3.2

For the forward KL divergence :

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

For the reverse KL divergence :

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

As we can see : the forward KL divergence only require $\boldsymbol{x}$ thus we can use it for the second scenario. The reverse KL divergence only require to evaluate p_X thus we can use it for the first scenario.

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 4.1

Gradient ascent on $V(d, g)$ w.r.t d :

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

Gradient descent on $V(d, g)$ w.r.t g :

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

Rewriting in matrix :

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

With :

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

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 4.2

A stationary point is a point on the surface of the graph where all its partial derivatives are zero (equivalently, the gradient is zero) :

$$\begin{aligned} \nabla V(d, g) &= \begin{pmatrix} \frac{\partial}{\partial d} V(d, g) \\ \frac{\partial}{\partial g} V(d, g) \end{pmatrix} \\ &= \begin{pmatrix} g \\ d \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Rightarrow \begin{cases} g = 0 \\ d = 0 \end{cases} \end{aligned}$$

Thus the stationary point is at $(g = 0, d = 0)$.

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 $k \to \infty$.

Answer 4.3

Let's find the eigenvalues of $\boldsymbol{A}$:

$$\begin{aligned} \det|\boldsymbol{A} - \lambda \boldsymbol{I}| = 0 \Rightarrow \det \left| \begin{pmatrix} 1 - \lambda & \alpha \\ -\alpha & 1 - \lambda \end{pmatrix} \right| &= 0 \\ \Rightarrow (1 - \lambda)^2 + \alpha^2 &= 0 \\ \Rightarrow \lambda_1 = 1 + i\alpha \quad \text{and} \quad \lambda_2 &= 1 - i\alpha \end{aligned}$$

As we can see $\boldsymbol{A}$ has complex eigenvalues, and we know that complex eigenvalues imply rotation. So the updates will be rotating in the parameter space and will never converge to a balance as $k \to 0$.

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:

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

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 is the average augmented view of a data point $\boldsymbol{x}$ 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 5.1

We have :

$$\begin{aligned}
\|W_p\boldsymbol{f}_1 - \text{Stop-Grad}(\boldsymbol{f}_2)\|_2^2 &= (W_p\boldsymbol{f}_1 - \boldsymbol{f}_2)^\top(W_p\boldsymbol{f}_1 - \boldsymbol{f}_2) \\
&= W_p^\top\boldsymbol{f}_1^\top W_p\boldsymbol{f}_1 - W_p^\top\boldsymbol{f}_1^\top\boldsymbol{f}_2 - \boldsymbol{f}_2^\top W_p\boldsymbol{f}_1 + \boldsymbol{f}_2^\top\boldsymbol{f}_2 \\
&= \text{tr}(W_p^\top W_p\boldsymbol{f}_1\boldsymbol{f}_1^\top) - \text{tr}(W_p^\top\boldsymbol{f}_2\boldsymbol{f}_1^\top) - \text{tr}(W_p\boldsymbol{f}_1\boldsymbol{f}_2^\top) + \text{tr}(\boldsymbol{f}_2\boldsymbol{f}_2^\top)
\end{aligned}$$

Thus we have :

$$\begin{aligned}
J(W, W_p) &= \frac{1}{2}\mathbb{E}_{x_1,x_2}\left[\|W_p\boldsymbol{f}_1 - \text{Stop-Grad}(\boldsymbol{f}_2)\|_2^2\right] \\
&= \frac{1}{2}\left[\text{tr}(W_p^\top W_p\mathbb{E}[\boldsymbol{f}_1\boldsymbol{f}_1^\top]) - \text{tr}(W_p\mathbb{E}[\boldsymbol{f}_1\boldsymbol{f}_2^\top]) - \text{tr}(\mathbb{E}[\boldsymbol{f}_2\boldsymbol{f}_1^\top]W_p) + \text{tr}(\mathbb{E}[\boldsymbol{f}_2\boldsymbol{f}_2^\top]))\right] \\
&= \frac{1}{2}\left[\text{tr}(W_p^\top W_pF_1) - \text{tr}(W_pF_{12}) - \text{tr}(F_{12}W_p) + \text{tr}(F_2))\right]
\end{aligned}$$

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).

Answer 5.2

Taking the partial derivative w.r.t W_p and we get the gradient update rule :

$$\begin{aligned}
\dot{W}_p &= -\frac{\partial J}{\partial W_p} \\
&= -W_pF_1 + F_{12}
\end{aligned}$$

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:

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

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

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

Answer 5.3

With $\tilde{W}_p(t) = (W_p(t) - I_{n_2})$, let's take the partial derivative w.r.t W :

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

Using : $\mathrm{vect}(AXB) = (B^\top \otimes A)\mathrm{vect}(X)$, a property that connects vectorization and the Kronecker product, we have :

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

So if the minimal eigenvalue $\lambda_{min}(H(t))$ is bounded away from zero : $\inf_{t\geq 0} \lambda_{min}(H(t)) \geq \lambda_0 > 0$, then by using the property above, we have : $\|W(t)\|_2 = e^{-\lambda_0 t}\|W(0)\|_2$. Thus $W(t) \to 0$.

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.

Answer 5.4

When both the Stop-Grad and the predictor are removed : $W_p = I$, replacing this in the partial derivative w.r.t W, we obtain a similar form as question 5.3 :

$$\dot{W} = -X'W$$

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.

By assuming that $X^{'}$ is a positive definite matrix, with a similar arguments as the question above, if $\inf_{t\geq 0} \lambda_{min}(X^{'}) \geq \lambda_0 > 0$, we have : $\|W(t)\|_2 = e^{-\lambda_0 t}\|W(0)\|_2$. Thus $W(t) \to 0$.

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

Answer 5.5

Removing the stop-gradient and the predictor yields a gradient update for the parameters $W(t)$, and as we show in the last questions : if the minimal eigenvalue over time is bounded away from zero then $W(t) \to 0$. Thus there is no chance for $W(t)$ to lean any meaningful features.