Instructions

- *For all questions, show your work!*
- *Use a document preparation system such as LaTeX.*
- *Submit your answers electronically via the course gradescope*
- *TA for this assignment is (theoretical part) :* ***Alexandra Volokhova*** *(IFT6135B) and* ***Ghait Boukachab*** *(IFT6135A).*

**Question 1** (2-2-4-2)**.** 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) :

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

We assume $q_\phi \in \mathcal{Q}$ where $\mathcal{Q}$ is a parametric family, where we use $\phi$ to specify which member of the family we are using.

1.1 Show that data log likelihood $\log p_\theta(\boldsymbol{x})$ can be decomposed as a sum of ELBO and KL-divergence between variational and true posteriors over $\boldsymbol{z}$ : $D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x}))$

1.2 Show that maximizing ELBO w.r.t. $\phi$ is equivalent to minimizing KL-divergence between variational and true posteriors over $\boldsymbol{z}$ w.r.t. $\phi$.

1.3 In this and following task, the goal is to compare armotized variational inference (when $q_\phi$ is optimised for the whole dataset) with the traditional variational inference (when $q_\phi$ is optimised individually for each $x$). Consider a finite training set $\{\boldsymbol{x}_i : i \in \{1, ..., n\}\}$, $n$ being the size the training data. Let's fix $\theta$ for simplicity. Let $q^* = \arg\max_{q_\phi \in \mathcal{Q}} \sum_{i=1}^{n} \mathcal{L}(\theta, \phi; \boldsymbol{x}_i)$ (i.e. $q^*$ is the optimal variational distribution in the family $\mathcal{Q}$ for given $\theta$ and training set). In addition, for each $\boldsymbol{x}_i$ let $q_i^* = \arg\max_{q_\phi \in \mathcal{Q}} \mathcal{L}(\theta, \phi; \boldsymbol{x}_i)$. Compare $D_{\mathrm{KL}}(q^*(\boldsymbol{z}|\boldsymbol{x}_i)||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i))$ and $D_{\mathrm{KL}}(q_i^*(\boldsymbol{z}|\boldsymbol{x}_i)||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i))$. Which one is bigger ?

1.4 Following the previous question, compare the two approaches in the second subquestion (justify the answers).
   (a) which approach is better for estimating marginal likelihood via empirical ELBO
   (b) which one is more computationally efficient
   (c) which one is more memory efficient (storage of parameters)

**Question 2** (5-2-7-2-2-5-2)**.** In this task, we will go deeper into mathematics of diffusion models. Consider a denosing diffusion probabilistic model (DDPM) with the encoder process given by a linear Gaussian model : $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t|\sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t I)$, where $\beta_t \in (0, 1)$ is a fixed noise schedule. The forward diffusion process starts from initial image $\boldsymbol{x}_0$ from the dataset and ends at $\boldsymbol{x}_T$ ($T$ is a fixed number of steps). We assume $\boldsymbol{x}_T \sim p(\boldsymbol{x}_T) = \mathcal{N}(\boldsymbol{x}_T|0, I)$. The goal of the training is to learn a reversed (denoising process) $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$, which will allow to generate an image $\boldsymbol{x}_0$ starting from Gaussian noise $\boldsymbol{x}_T$.

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.

2.1 Given the equation for linear Gauassian encoder process $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, show that the ground truth denoising process is

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_{t-1}|\tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0), \tilde{\beta}_t I) \tag{1}$$

where

$$\tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t \tag{2}$$

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

$$\alpha_t = 1-\beta_t$$

$$\bar{\alpha}_t = \prod_{s=1}^{t}\alpha_s$$

If needed, you can use the following equation without proving it :

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t|\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)I) \tag{3}$$

Hint : use Bayes rule and Markovian property of the encoder process

2.2 As we saw in task 2.1, it is possible to reverse the diffusion process analytically, without training anything. Explain, why we still need to train the reverse process $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ to generate images.

2.3 Now, let's derive the objective function for DDPM. Essentially, DDPM is a hierarchical variational autoencoder (with latent variables $\{x_1, \ldots, x_T\}$) and its objective is an evidence lover bound (ELBO) for $\log p(x_0)$. Show that

$$\log p(x_0) \geq \mathcal{L}_{DDPM}(\theta; \boldsymbol{x}_0) = -L_0(\boldsymbol{x}_0) - \sum_{t=2}^{T} L_{t-1}(\boldsymbol{x}_0) - L_T(\boldsymbol{x}_0)$$

where

- (reconstruction term) $L_0(\boldsymbol{x}_0) = -\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)} \log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)$
- (denoising matching term) $L_{t-1}(\boldsymbol{x}_0) = \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))$
- (prior matching term) $L_T(\boldsymbol{x}_0) = D_{\mathrm{KL}}(q(\boldsymbol{x}_T|\boldsymbol{x}_0)||p(\boldsymbol{x}_T))$

The following equations migth be useful for derivations :

$$p_\theta(\boldsymbol{x}_{0:T}) = p(\boldsymbol{x}_T)\prod_{t=1}^{T} p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$$

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_0) = \frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)q(\boldsymbol{x}_t|\boldsymbol{x}_0)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}$$

2.4 Which term in $\mathcal{L}_{DDPM}$ doesn't affect optimisation over parameters and therefore can be excluded from the objective function ?

2.5 Compare ELBO for vanila VAE (see previous task) and ELBO for DDPM. What is the key difference between them (in terms of trainable parameters) ?

2.6 Let's consider $L_{t-1}(\boldsymbol{x}_0)$ and $L_0(x_0)$ in more detail.

- Using eq. 2 and 3, show that $\tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right)$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}|0, I)$

- A common parametrisation for denoising process is $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}|\mu_\theta(\boldsymbol{x}_t, t), \sigma_t^2 I)$, where mean of the Gaussian $\mu_\theta(\boldsymbol{x}_t, t)$ is trainable (here we consider $\sigma_t^2$ to be fixed for simplicity, while in practice it is trainable). However, instead of training a model to predict the $\mu_\theta(\boldsymbol{x}_t, t)$ directly, a common choice is to train a neural network $\boldsymbol{\epsilon}_\theta$ (aka "denoiser") to predict only the noise term : $\mu_\theta(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\big(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\big)$. Show that

$$\mathbb{E}_{q(\boldsymbol{x}_0)} L_{t-1}(\boldsymbol{x}_0) = \mathbb{E}_{q(\boldsymbol{x}_0), \boldsymbol{\epsilon} \sim \mathcal{N}(0,I)} \left[\lambda_t ||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\big(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\big)||^2\right] + const \tag{4}$$

  where $\lambda_t = \frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1-\bar{\alpha}_t)}$ and $q(\boldsymbol{x}_0)$ is the groundtruth data distribution
  Hint : you can use the equation for KL divergence between multivariate normal distributions without deriving it.
- Show that $\mathbb{E}_{q(\boldsymbol{x}_0)} L_0(\boldsymbol{x}_0)$ can be written in the same way as eq. 4

2.7 Finally, put together the equations for ELBO terms and get the DDPM loss function.

**Question 3** (3-7)**.** Let $p_0$ and $p_1$ be two probability distributions with densities $f_0$ and $f_1$ (respectively). We want to explore what we can do with a trained GAN discriminator. A trained discriminator is thought to be one which is "close" to the optimal one :

$$D^* := \arg\max_D \mathbb{E}_{\boldsymbol{x} \sim p_1}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x} \sim p_0}[\log(1 - D(\boldsymbol{x}))].$$

3.1 For the first part of this problem, derive an expression we can use to estimate the Jensen-Shannon divergence using a trained discriminator. We remind that the definition of JSD is $\mathrm{JSD}(p_0, p_1) = \frac{1}{2}\big(KL(p_0 \| \mu) + KL(p_1 \| \mu)\big)$, where $\mu = \frac{1}{2}(p_0 + p_1)$.

3.2 For the second part, we want to demonstrate that a optimal GAN Discriminator (i.e. one which is able to distinguish between examples from $p_0$ and $p_1$ with minimal NLL loss) can be used to express the probability density of a datapoint $\boldsymbol{x}$ under $f_1$, $f_1(\boldsymbol{x})$ in terms of $f_0(\boldsymbol{x})$ [2]. Assume $f_0$ and $f_1$ have the same support. Show that $f_1(\boldsymbol{x})$ can be estimated by $f_0(\boldsymbol{x})D(\boldsymbol{x})/(1 - D(\boldsymbol{x}))$ by establishing the identity $f_1(\boldsymbol{x}) = f_0(\boldsymbol{x})D^*(\boldsymbol{x})/(1 - D^*(\boldsymbol{x}))$.

*Hint : Find the closed form solution for $D^*$.*

**Question 4** (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 :

$$J(W, W_p) = \frac{1}{2}\mathbb{E}_{x_1, x_2}\left[\|W_p \boldsymbol{f}_1 - \text{Stop-Grad}(\boldsymbol{f}_2)\|_2^2\right]. \tag{5}$$

2. You might need to use the “functional derivative” to solve this problem. See “19.4.2 Calculus of Variations” of the Deep Learning book or “Appendix D Calculus of Variations” of Bishop's Pattern Recognition and Machine Learning for more information.

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

$$J(W, W_p) = \frac{1}{2}\left[\mathrm{tr}(W_p^{\mathrm{T}} W_p F_1) - \mathrm{tr}(W_p F_{12}) - \mathrm{tr}(F_{12} W_p) + \mathrm{tr}(F_2))\right], \tag{6}$$

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 [3].

4.2 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 $W_p$).

4.3 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 [4], $\tilde{W}_p(t) = (W_p(t) - I_{n_2})$, and "vec(W)" refers to the *vectorization* of a matrix W [5]. For simplicity, we are not taking weight decay into account here [6].

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

4.4 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.

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

3. https ://en.wikipedia.org/wiki/Trace_(linear_algebra)https ://en.wikipedia.org/wiki/Trace_(linear_algebra).
4. For more information, see https ://en.wikipedia.org/wiki/Kronecker_product#Matrix_equationshttps ://en.wikipedia.org/
5. Also known as the "vec trick", it is obtained by stacking all the columns of a matrix A into a single vector.
6. 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.