Homework 5

Instructions: Complete the following questions. There are a total of 29 points for this homework. Each question is marked with its corresponding points. We will be using this notebook. Please include any code you write in your report or simply convert your notebook to a PDF and submit that alongside your report.

Collaboration: If you collaborate with other students on the homework, list the names of all your collaborators.

Notation: We will use this set of math notation specified on course website. For example, c is a scalar, $\mathbf{b}$ is a vector and $\mathbf{W}$ is a matrix.
You are encouraged (although not enforced) to follow this notation.

Variational Autoencoders (14pt)

Variational autoencoders (VAEs), unlike standard autoencoders, are generative models. We would like to sample a vector from some standard normal distribution $p(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and have that sample map to some element in the data distribution $p(\mathbf{x})$.

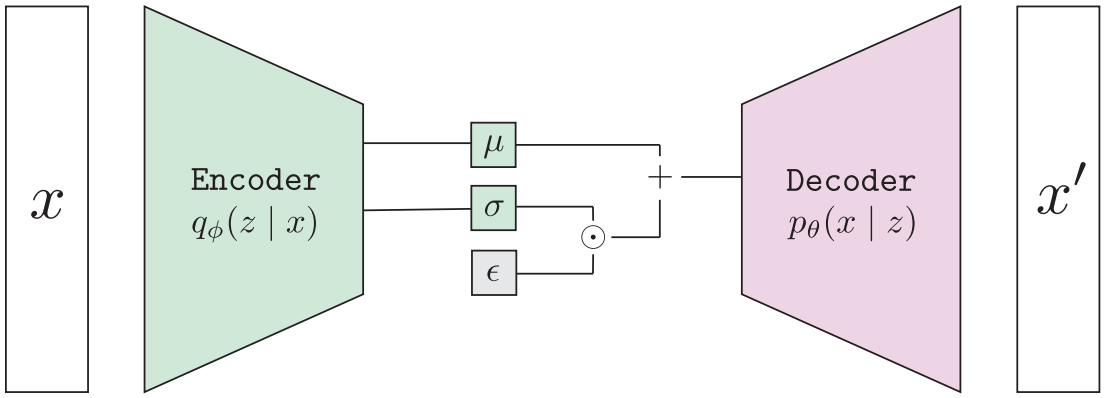

Figure 1: A variational autoencoder (VAE) diagram

Instead of directly maximizing $p(\mathbf{x})$, we instead aim to maximize a lower bound on $p(\mathbf{x})$

Homework 5

called the evidence lower bound (ELBO) Kingma and Welling [2019]:

$$\mathcal{L}_{\text{ELBO}} = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]}_{\text{first term}} - \underbrace{D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z}))}_{\text{second term}}, \tag{1}$$

where $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a fixed prior distribution over the latent $\mathbf{z}$.

1. **(1 pt)** Explain what each of each of the two terms in the above loss function (i.e., the first term and the second term) is doing. Please answer in a few sentences. Note that here we aim to *maximize* the loss function.

2. **(1 pt)** In a VAE, each input datapoint x is mapped to a distribution in the latent space by encoding each x as both a mean μ and variance σ^2:

$$\mu_z = f_\phi^\mu(x),$$
$$\sigma_z^2 = f_\phi^\sigma(x).$$

   In this question, we will think about how rich the structure of the means in the embedding space (i.e., the quantities $f_\phi^\mu(x)$ for all input datapoints) alone must be in a fully-trained VAE. Suppose you successfully train a VAE such that $q_\phi(\mathbf{z})$ becomes a unit Gaussian, and $p_\theta(x)$ is equal to the data distribution p_{data}. In other words, assume you achieved *perfect* encoder and decoder after training a VAE.

   (a) **(0.5 pt)** Does this imply that decoding only the means of the embeddings recovers the data distribution? In other words, are the sets of means of embedding sufficient for data distribution recovery, or are the variances of embeddings also necessary? Explain your answer in a few sentences.

   (b) **(0.5 pt)** Does this imply that the means of the embeddings (i.e., $f_\phi^\mu(x)$ above) are Gaussian distributed? In other words, is the random variable $f_\phi^\mu(x)$ Gaussian when $x \sim p_{\text{data}}$? Explain your answer in a few sentences.

   Hint: Note that the random variable $\mathbf{z} = \mathcal{N}(f_\phi^\mu(x), f_\phi^\sigma(x))$ is unit Gaussian by assumption, when $x \sim p_{\text{data}}$. Does this imply that its mean is also Gaussian?

3. **(1 pt)** Suppose you successfully train an autoencoder (not VAE) such that $g(f(\mathbf{x})) = \mathbf{x}$ where f is the deterministic encoder and g is the deterministic decoder. Does this imply that $g(\mathbf{z}) \sim p_{\texttt{data}}$ when $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$? Explain your answer in a few sentences.

4. **(6 pts)** Implement the VAE architecture in the provided colab.

5. **(1 pt)** Train the VAE model on the provided FashionMNIST dataset.

6. **(4 pt)** Using your trained model, complete the `plot_latents` function such that for a given pair of latent dimensions (i, j) the `plot_latents` plots a 10×10 grid of images sampled from different pairs of latent values in dimensions i, j (Figure 2). Latent dimensions not equal to i or j should be set to zero.

Homework 5

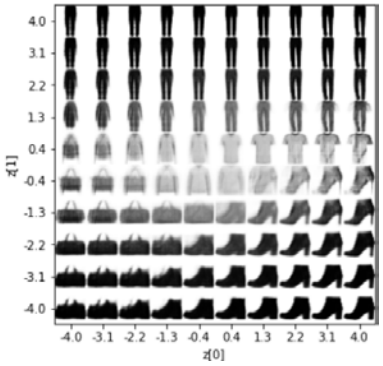

Figure 2: Example latent visualization.

Diffusion Models (15pt)

A diffusion process works by sequentially adding noise to a given input x_0 for T timesteps, and our goal is to learn a function that removes the added noise (i.e., *denoising*). While there are many variations around this central concept, we will be looking closely at one particular approach from [Ho et al., 2020].

As T approaches infinity, this process results in noise drawn from a zero-mean isotropic Gaussian at $\mathbf{x}_T$. This process is depicted in Figure 3. The random variable $\mathbf{x}_t$ conditioned on $\mathbf{x}_{t-1}$ is distributed according to $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ with the following form:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\sqrt{\alpha_t}\,\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \quad \text{where} \quad \alpha_t = 1 - \beta_t. \tag{2}$$

Note: the distribution q refers to the joint distribution over over all $\mathbf{x}_0, \ldots, \mathbf{x}_T$. Conditionals of this distribution are denoted as $q(\cdot|\cdot)$ for subsets of the variables in the joint configuration.

Let's first try to understand the properties of the forward diffusion process in terms of $q(\mathbf{x}_t|\mathbf{x}_0)$. We will use these properties later to derive a loss for the reverse diffusion process.

7. **(2pt)** Adding two independent Gaussian random variables results in another Gaussian random variable. Using this property, one can derive a closed-form representation for the Gaussian distribution $q(\mathbf{x}_t|\mathbf{x}_0)$ in terms of α_t and $\mathbf{x}_0$. Prove that

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

   where $\bar{\alpha}_t$ is referred to as $\prod_{s=1}^{t} \alpha_s$. Provide a detailed explanation of how you arrived at your answer.

8. **(1pt)** Let us calculate how far $\mathbf{x}_T$ is from being $\mathcal{N}(\mathbf{0}, \mathbf{I})$ in KL divergence. For any d-dimensional Gaussian distribution $\mathcal{N}(\mathbf{a}, \Sigma)$, we have

   $$D_{\mathrm{KL}}\big(\mathcal{N}(\mathbf{a}, \Sigma) \parallel \mathcal{N}(\mathbf{0}, \mathbf{I})\big) = \frac{1}{2}\left(\|\mathbf{a}\|_2^2 + \mathrm{tr}(\Sigma) - \log(\det(\Sigma)) - d\right),$$

where $\mathrm{tr}(\Sigma)$ is the sum of diagonal entries of the covariance matrix Σ. Use the above formula and compute $D_{\mathrm{KL}}\big(q(\mathbf{x}_T|\mathbf{x}_0) \parallel \mathcal{N}(\mathbf{0}, \mathbf{I})\big)$. Conclude that if T is large, then $q(\mathbf{x}_T|\mathbf{x}_0)$ is close to being unit Gaussian.

Now, our real goal is to learn a function $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mu_\theta(\mathbf{x}_t), \Sigma_t)$ that removes the noise added (i.e., *denoising*). Indeed, learning such a function enables us to remove noise from $\mathbf{x}_t$ to recover $\mathbf{x}_{t-1}$. By starting with a completely noisy image (i.e., a unit Gaussian), we can perform sequential denoising to generate an image with this approach.

One strategy to learning $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is to minimize the KL-divergence between this distribution and the true reverse process $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$:

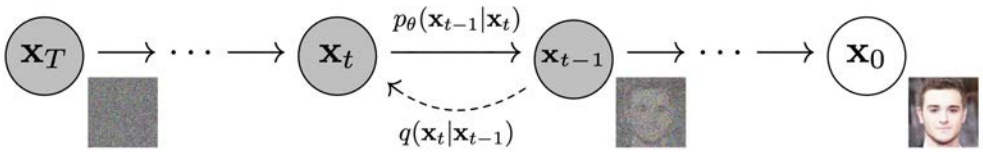

Figure 3: The directed graphical model for the diffusion process.

But what is the form of the reverse diffusion process $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$? We will derive this in the following questions.

To derive this expression, we use the closed form for the expression $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$, which works out to be (you may refer to Ho et al. [2020] for more details):

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\tilde{\mu}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}) \tag{4}$$

$$\tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \cdot \beta_t \tag{5}$$

$$\tilde{\mu}(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \cdot \mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t} \cdot \mathbf{x}_0 \tag{6}$$

Also, note from Eqn. 3, we have that

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{(1 - \bar{\alpha}_t)}\epsilon_t, \tag{7}$$

where $\epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a Gaussian random variable with unit variance.

Using the above expressions:

9. **(1pt)** Derive an expression for mean and the variance of $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ in terms of the above quantities.

Homework 5

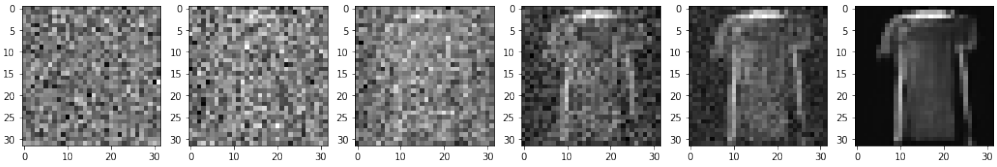

Figure 4: Diffusion inference figure example

10. **(1pt)** Derive an expression for $D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))$.

    Hint: You can use the following formula for the KL divergence between two d-dimensional Gaussian vectors:

$$D_{\mathrm{KL}}\left(\mathcal{N}\left(\mathbf{a}, \Sigma_a\right) \parallel \mathcal{N}\left(\mathbf{b}, \Sigma_b\right)\right) = \frac{1}{2}\Big(\log(\det(\Sigma_b)) - \log(\det(\Sigma_a)) - d + (\mathbf{a}-\mathbf{b})^T\Sigma_b^{-1}(\mathbf{a}-\mathbf{b}) + \mathrm{tr}(\Sigma_b^{-1}\Sigma_a)\Big).$$

Congrats! You now have an objective function for denoising, which you can use to train a diffusion model.

In practice, however, people often reparameterize the optimization problem in terms of $\epsilon_t = \mathbf{x}_t - \mathbf{x}_{t-1}$ and predict ϵ_t with a model $\epsilon_\theta(\mathbf{x}_t, t)$. Doing this, and other simplifications (see Ho et al. [2020] for details), we get the following diffusion modeling loss, which is used in the coding section of this pset:

$$\mathcal{L}_{\texttt{Diffusion}} = \|\epsilon_t - \epsilon_\theta(\mathbf{x}_t, t)\|_2^2. \tag{8}$$

11. **(6pt)** In the provided colab, complete the `FIXME`s and train the diffusion model.

12. **(2pt)** Create a plot with six horizontally-aligned subfigures showing the diffusion model inference $\mathbf{x}_t$ for $t \in \{200, 100, 50, 20, 10, 0\}$. An example can be seen in Figure 4. Your plots will look different.

13. **(2pt)** There are many similarities between VAEs and diffusion models. Answer each question with a sentence or two.

    (a) **(1pt)** What is the equivalent of the encoder, decoder, and latent variable in the diffusion model?

    (b) **(1pt)** What is the difference between the encoder in a diffusion model and VAE?

References

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020. URL https://arxiv.org/abs/2006.11239.

Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *CoRR*, abs/1906.02691, 2019. URL http://arxiv.org/abs/1906.02691.

6.7960 Deep Learning

Fall 2024