Homework 5

Instructions: Complete the following questions. There are a total of 32 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.

Submission: Upload a **PDF** of your response through **Canvas** by **11/14 at 1pm**.

Notation: We will use this set of math notation specified on course website, whose LaTeX source is available on Canvas. 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 a 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 $p(\mathbf{x})$.

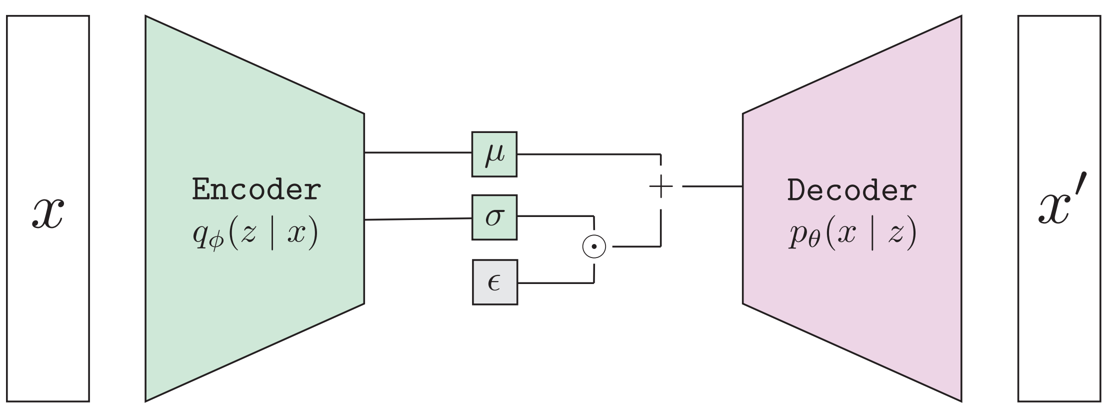

Figure 1: A variational autoencoder (VAE) diagram

Instead of directly maximizing $p(\mathbf{x})$, we instead aim to maximize a lower bound on $p(\mathbf{x})$ called the evidence lower bound (ELBO) Kingma and Welling [2019].

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

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 are doing. Please answer in a few sentences.

2. **(1 pt)** Suppose you successfully train a VAE such that $q_\phi(\mathbf{z})$ becomes a unit Gaussian. Does this imply that the means of the embeddings ($f_\phi^\mu(\mathbf{x})$ from lecture notes) are Gaussian distributed? Explain your answer in a few sentences.

3. **(1 pt)** Suppose you successfully train an autoencoder (not VAE) such that $g(f(\mathbf{x})) = \mathbf{x}$ where f is the determinisic 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. Latent dimensions not equal to i or j should be set to zero.

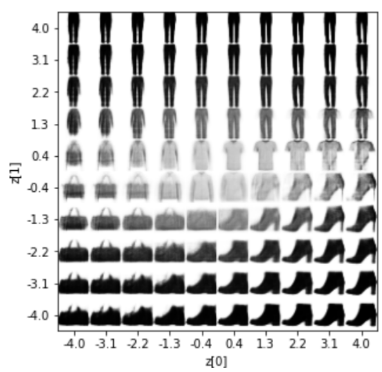

Figure 2: Example latent visualization

Diffusion Models (16pt)

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. 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:

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

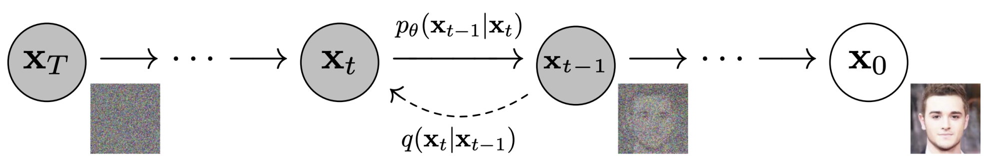

Figure 3: The directed graphical model for the diffusion process

Our 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 when sampling from $q(\mathbf{x}_t|\mathbf{x}_{t-1})$. One simple way of training this is to sample a batch of $(\mathbf{x}_0, t)$ tuples from your dataset, simulate noise up to timestep t for each data point, and train your model with the loss $||\mu_\theta(\mathbf{x}_t) - \mathbf{x}_{t-1}||_2$.

Unfortunately, with T often in the thousands, simulating added noise incrementally for each datapoint is computationally burdensome. Instead, we want a closed-form representation for $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ so that we can directly minimize the KL divergence between $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ and $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ without needing to simulate thousands of steps.

7. **(2pt)** Adding two independent Gaussian random variables results in another Gaussian random variable. Using this property, derive a closed form representation of the form $\mathcal{N}(\cdot, \cdot)$ for the Gaussian distribution $q(\mathbf{x}_t|\mathbf{x}_0)$ in terms of α_t and $\mathbf{x}_0$. Show how you arrived at your answer. You may also use $\bar{\alpha}_t$ to refer to $\prod_{s=1}^{t} \alpha_s$.

8. **(2pt)** Given an $\epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, write the random variable $\mathbf{x}_0$ in terms of $\epsilon_t, \bar{\alpha}_t$ and $\mathbf{x}_t$.

Now that we have $\mathbf{x}_0$ in terms of $\mathbf{x}_t$, we will find a closed-form expression for $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ and substitute our above expression in for $\mathbf{x}_0$ to find $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$. The algebra for this part is a bit tricky, so we provide you with $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$. See [Ho et al., 2020] for details.

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\tilde{\mu}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}) \tag{3}$$

$$\tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \cdot \beta_t \tag{4}$$

$$\tilde{\mu}(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \cdot \mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t} \cdot \mathbf{x}_0 \tag{5}$$

9. **(2pt)** Plug your expression for $\mathbf{x}_0$ into $\tilde{\mu}(\mathbf{x}_t, \mathbf{x}_0)$ and simplify to obtain $\tilde{\mu}(\mathbf{x}_t)$ in terms of $\epsilon_t, \alpha_t, \bar{\alpha}_t$, and $\mathbf{x}_t$.

Our goal is to minimize $D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))$, and we can do so by minimizing the difference between $\tilde{\mu}(\mathbf{x}_t, \mathbf{x}_0)$ and $\mu_\theta(\mathbf{x}_t)$. An addition, since $\mathbf{x}_t$ is known during inference, finding θ that minimizes the difference between $\tilde{\mu}(\mathbf{x}_t, \mathbf{x}_0)$ and $\mu_\theta(\mathbf{x}_t)$ reduces to the simpler problem of finding θ that minimizes the following loss

$$\mathcal{L}_{\texttt{Diffusion}} = ||\epsilon_t - \epsilon_\theta(\mathbf{x}_t)||_2. \tag{6}$$

10. **(6pt)** In the provided colab, complete the `FIXME`s and train the diffusion model.

11. **(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 11. Your plots will look different.

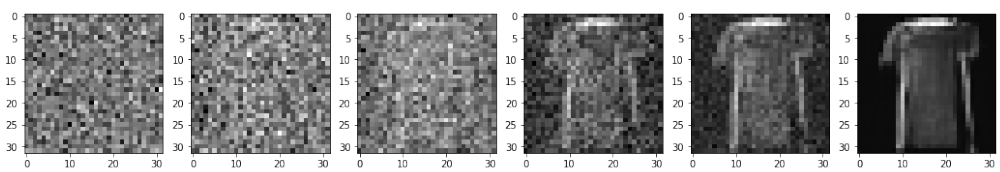

Figure 4: Diffusion inference figure example

12. **(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 a 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`.