Homework 4:
Deep Generative Model

Machine Learning and Data Mining (Fall 2023)

Student Name: Student ID:

Lectured by: Shangsong Liang
Sun Yat-sen University

Your assignment should be submitted to the email that will be provided by the TA
Deadline of your submission is: 23:59PM, January 05, 2024
Do NOT Distributed This Document and the Associated Datasets

Problem: Implementing the Variational Autoencoder (VAE)

For this problem, we will be using PyTorch to implement the variational autoencoder (VAE) and learn a probabilistic model of the MNIST dataset of handwritten digits. Formally, we observe a sequence of binary pixels $x \in \{0,1\}^d$, and let $z \in \mathbb{R}^k$ denote a set of latent variables. Our goal is to learn a latent variable model $p_\theta(x)$ of the high-dimensional data distribution $p_{\text{data}}(x)$.
The VAE is a latent variable model that learns a specific parameterization $p_\theta(x) = \int p_\theta(x,z)\,dz = \int p(z)p_\theta(x|z)\,dz$. Specifically, the VAE is defined by the following generative process:

$$p(z) = \mathcal{N}(z|0,I)$$
$$p_\theta(x|z) = \text{Bernoulli}(x|f_\theta(z))$$

In other words, we assume that the latent variables z are sampled from a unit Gaussian distribution $\mathcal{N}(z|0,I)$. The latent z are then passed through a neural network decoder $f_\theta(\cdot)$ to obtain the parameters of the d Bernoulli random variables which model the pixels in each image.
Although we would like to maximize the marginal likelihood $p_\theta(x)$, computation of $p_\theta(x) = \int p(z)p_\theta(x|z)\,dz$ is generally intractable as it involves integration over all possible values of z. Therefore, we posit a variational approximation to the true posterior and perform amortized inference as we have seen in class:

$$q_\phi(z|x) = \mathcal{N}(z|\mu_\phi(x), \text{diag}(\sigma^2_\phi(x)))$$

Specifically, we pass each image x through a neural network which outputs the mean μ_ϕ and diagonal covariance $\text{diag}(\sigma^2_\phi(x))$ of the multivariate Gaussian distribution that approximates the distribution over latent variables z given x. We then maximize the lower bound to the marginal log-likelihood to obtain an expression known as the evidence lower bound (ELBO):

$$\log p_\theta(x) \geq \text{ELBO}(x;\theta;\phi) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \text{D}_{KL}(q_\phi(z|x)\,||\,p(z))$$

Notice that the ELBO, as shown on the right-hand side of the above expression, decomposes into two terms: (1) the reconstruction loss: $-\mathbb{E}_{q_\phi(z)}[\log p_\theta(x|z)]$, and (2) the Kullback-Leibler (KL) term: $\text{D}_{KL}(q_\phi(z|x)\,||\,p(z))$.

Your objective is to implement the variational autoencoder by modifying `utils.py` and `vae.py`.

1. Implement the reparameterization trick in the function `sample_gaussian` of `utils.py`. Specifically, your answer will take in the mean μ and variance σ^2 of the Gaussian distribution $q_\phi(z|x)$ and return a sample $z \sim q_\phi(z|x)$.

2. Next, implement negative ELBO bound in the file `vae.py`. Several of the functions in `utils.py` will be helpful, so please check what is provided. Note that we ask for the negative ELBO, as PyTorch optimizers minimize the loss function. Additionally, since we are computing the negative ELBO

over a mini-batch of data $\{x^{(i)}\}_{i=1}^n$, make sure to compute the average $-\frac{1}{n}\sum_{i=1}^n \text{ELBO}(x^{(i)};\theta;\phi)$ over the minibatch. Finally, note that the ELBO itself cannot be computed exactly since exact computation of the reconstruction term is intractable. Instead, we ask that you estimate the reconstruction term via Monte Carlo sampling:

$$-\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] \approx -\log p_\theta(x|z^{(1)})$$

here $z^{(1)} \sim q_\phi(z|x)$ denotes a single sample. The function `kl_normal` in `utils.py` will be helpful. Note: negative ELBO bound also expects you to return the average reconstruction loss and KL divergence.

3. To test your implementation, run `python run_vae.py` to train the VAE. Once the run is complete (20000 iterations), it will output (assuming your implementation is correct): the average (1) negative ELBO, (2) KL term, and (3) reconstruction loss as evaluated on a test subset that we have selected. Report the three numbers you obtain as part of the write-up. Since we' re using stochastic optimization, you may wish to run the model multiple times and report each metric' s mean and corresponding standard error. (Hint: the negative ELBO on the test subset should be somewhere around 100.)

4. Visualize 200 digits (generate a single image tiled in a grid of 10×20 digits) sampled from $p_\theta(x)$.