CS 598-DGM: Spring'25
Deep Generative Models

Homework 1

(Due Monday, March 03, 11:59 pm)

- The homework is due at 11:59 pm on the due date. We will be using Gradescope for the homework assignments. **Please do NOT email a copy of your solution.** Contact the TAs (Zhijie, Rohan) if you are having technical difficulties in submitting the assignment. We will NOT accept late submissions.
- Please make sure that each question is clearly marked. You may use as many pages as needed but do not change the order of the questions and answers.
- **When submitting on Gradescope, you are required to assign the correct pages for each sub-problem to the provided outline. If pages are incorrectly assigned or left unassigned on Gradescope, it will result in no credit, and regrade requests regarding this will be declined.** Double-check your submission to ensure accuracy.
- Please use Slack first if you have questions about the homework. You can also come to our (zoom) office hours and/or send us e-mails. If you are sending us emails with questions on the homework, please start subject with "CS 598-DGM: " and send the email to *all course staff*: Arindam, Zhijie, and Rohan.
- The homework consists of both written assignments and programming assignments. Please submit your report as a PDF file and your code/model following the below instruction.

  Programming Assignment Instructions

  - All programming needs to be in Python 3.
  - The homework will be graded using Gradescope. You will be able to submit your code as many times as you want.
  - We provided a starter Jupyter Notebook on Canvas. Please finish the programming assignment based on it.
  - Please do not change the seed in the starter code.
  - For submitting on Gradescope, you need to upload four files: a Jupyter Notebook and three model checkpoints. Please make sure they are named in this way:
    1. "hw1.ipynb"
    2. "model_rbm_seed2025.pt"
    3. "model_vae_seed2025.pt"
    4. "model_vae2_seed2025.pt"
  - Please write your code entirely by yourself.

1. (20 points) Consider a Boltzmann machine over visible units $\mathbf{x} \in \{0,1\}^D$ and hidden units $\mathbf{z} \in \{0,1\}^d$ having energy function:

$$E(\mathbf{x}, \mathbf{z}; \theta) = -\mathbf{x}^T W \mathbf{z} - \frac{1}{2}\mathbf{x}^T B \mathbf{x} - \frac{1}{2}\mathbf{z}^T A \mathbf{z} \ , \tag{1}$$

where A, B are symmetric matrices and the diagonal elements of A and B are 0.

The corresponding distribution $p(\mathbf{x}; \theta) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}; \theta)$ with $p(\mathbf{x}, \mathbf{z}; \theta) = \frac{e^{-E(\mathbf{x},\mathbf{z};\theta)}}{Z(\theta)}$ is intractable because of the partition function $Z(\theta)$.

(a) (5 points) Consider any conditional distribution $q(\mathbf{z}|\mathbf{x}; \phi)$ with parameters ϕ and show how you will construct a variational lower bound to the log-likelihood $\log p(\mathbf{x}; \theta)$ as:

$$\log p(\mathbf{x}; \theta) \geq \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x};\phi)}[\log p(\mathbf{x}, \mathbf{z}; \theta)] + H(q) \ , \tag{2}$$

where $H(q)$ is the entropy of $q(\mathbf{z}|\mathbf{x}; \phi)$, with log denoting natural logarithm.

(b) (10 points) Consider the following specific form for q:

$$q(\mathbf{z}|\mathbf{x}; \phi) = \prod_{j=1}^{d} q(\mathbf{z}_j|\mathbf{x}; \phi) \tag{3}$$

where the unknown parameters ϕ are $\mu_j = q(\mathbf{z}_j = 1|\mathbf{x}), j \in [d]$. Show that the optimum set of variational parameters $\phi = \{\mu_j, j \in [d]\}$ which maximize the variational lower bound in (2) are given by the solution of the following system of fixed point equations:

$$\mu_j = \sigma\left(\sum_i W_{ij} x_i + \sum_{\ell \neq j} A_{\ell j} \mu_\ell \right) \ , \quad j \in [d] \ , \tag{4}$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function.

(c) (5 points) For any given matrices W, A, consider solving the fixed point equation in (4) by initializing $\mu_j^{(0)} = 0.5$ and doing iterative updates:

$$\mu_j^{(t+1)} = \sigma\left(\sum_i W_{ij} x_i + \sum_{\ell \neq j} A_{\ell j} \mu_\ell^{(t)} \right) \ , \quad j \in [d] \ . \tag{5}$$

Do you think the iterative updates will always converge? If you claim "yes," you have to give a convergence proof. If you claim "no," you have to give a clear argument or a counter-example.[1]

[1] It is ok, in fact advisable, to run code (say with small d, D) to get a sense of the iteration dynamics to get intuition for the proof or to help find a counter-example.

2. (25 points) Consider the problem of computing the following gradient:

$$\nabla_\mu \ \mathbb{E}_{z \sim N(\mu, \mathbb{I}_{d\times d})} \left[\frac{1}{2} \|z\|_2^2 \right] \ , \tag{6}$$

where $z, \mu \in \mathbb{R}^d$.

(a) (10 points) Consider estimating the gradient using the reparameterization approach based on $\epsilon \sim N(0, I_{d\times d})$, which does not depend on the parameter μ.

i. (3 points) Outline the algorithm for estimating the gradient based on L samples from $N(0, \mathbb{I}_{d\times d})$.

ii. (3 points) Prove that the expectation (w.r.t. the true distribution) of the single sample estimate of the gradient using reparameterization is μ.

iii. (4 points) Prove that the variance (w.r.t. the true distribution) of the single sample estimate of the gradient using reparameterization is d.

(b) (12 points) Consider estimating the gradient using the REINFORCE approach.

i. (3 points) Outline the algorithm for estimating the gradient based on L samples from $N(\mu, \mathbb{I}_{d\times d})$.

ii. (3 points) Prove that the expectation (w.r.t. the true distribution) of the single sample estimate of the gradient using REINFORCE is μ.

iii. (6 points) Prove that the variance (w.r.t. the true distribution) of the single sample estimate of the gradient using REINFORCE is $\Omega(d^3)$.

(c) (3 points) Professor Cool Friedrich Guess claims that

$$\nabla_\mu \ \mathbb{E}_{z \sim N(\mu, \mathbb{I}_{d\times d})} \left[\frac{1}{2} \|z\|_2^2 \right] = \mathbb{E}_{\mu \sim N(\mu, \mathbb{I}_{d\times d})} \left[\nabla_z \ \frac{1}{2} \|z\|_2^2 \right] = \mathbb{E}_{z \sim N(\mu, \mathbb{I}_{d\times d})} [z] = \mu \ . \tag{7}$$

Is Professor Guess correct? If yes, show why. If no, argue why not. In either case, a reference to a paper with technical argument proving/disproving the claim will be sufficient.

3. (15 points) Consider the Bayesian Linear Regression (BLR) model:

$$y = \beta^\top \mathbf{x} + \epsilon_i \ , \quad \beta \sim N(0, \mathbb{I}_{d\times d}) \ , \quad \epsilon \sim N(0, \sigma^2) \ . \tag{8}$$

Given an i.i.d. training set $(\mathbf{x}_i, y_i), i \in [n]$, we focus on a mean-field variational inference (MFVI) approach to estimating the unknown parameter σ in BLR. Consider the following variation distribution for MFVI:

$$q_\phi(\beta) = \prod_{j=1}^{d} q_{\phi_j}(\beta_j) = \prod_{j=1}^{d} N(\beta_j|\mu_j, s_j^2) \tag{9}$$

with variational parameters $(\mu_j, s_j^2), j \in [d]$.

(a) (7 points) Write down the ELBO for this MFVI only in terms of the variational parameters $(\mu_j, s_j^2), j \in [d]$, model parameter σ, and data $(\mathbf{x}_i, y_i), i \in [n]$. In particular, the ELBO expression should not have any expectations over any random variables.

(b) (8 points) Assuming that $(\sigma^t, \mu_j^t, s_j^t, j \in [d])$ are the parameter values from iteration t, show the gradient descent based update for each of these parameters to obtain $(\sigma^{t+1}, \mu_j^{t+1}, s_j^{t+1}, j \in [d])$.

4. (40 points) The programming assignment will focus on developing your own code for: Restricted Boltzmann Machine (RBMs) and Variational Auto-Encoders (VAEs).

   Dataset: The models will be evaluated on the (binary) MNIST dataset. We have provided code to download the dataset in the starter Jupyter notebook.

   Generative Models. We will consider three different generative models

   - (15 points) Restricted Boltzmann Machine (RBM): You will be implementing RBM learning based on Contrastive Divergence with k steps (CD-k) on the binary MNIST dataset. The dimensionality d of the latent variables $\mathbf{z} \in \{0,1\}^d$ is 128. You need to do the following:
     - Implement the RBM with CD-k based on the starter code. We will auto-grade your code by evaluating the trained model on the test set based on free energy.
     - Try three different numbers of steps $k = 1, 5, 25$ in CD-k. Plot sampled images and briefly discuss the effect of k in your report.
   - (10 points) Variational Auto-Encoder with MLP (VAE1): You will be implementing the VAE with Gaussian latent variables on the MNIST dataset. The encoder and decoder are MLPs with one hidden layer (`hidden_dim=500`) and `tanh` as the non-linear activation function. Train the model by minimizing $-\text{ELBO} = -\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] + D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}))$. You can simply use the Mean Square Error for the reconstruction loss (it is the exact NLL (negative log-likelihood) up to a constant when $\mathbf{x}$ has a Gaussian distribution). You need to do the following:
     - Implement the VAE1 based on the starter code. We will auto-grade your code by evaluating the trained model based on the test set reconstruction loss.
     - Try four different latent dimension $N_z = 2, 5, 10, 20$. Plot sampled images and briefly discuss the effect of the choice of N_z in your report.
   - (15 points) Variational Auto-Encoder with CNN (VAE2): You will be implementing the VAE with Gaussian latent variables on the MNIST dataset. Let N_z denote the dimensionality of the latent variable z. The encoder is a CNN with
     - 3 Convolution-ReLU-MaxPool layers (Table 1), followed by
     - a linear layer mapping to N_z dimensions.

     The decoder is based on deconvolution with

     - a linear layer mapping from N_z dimensions, followed by
     - 3 DeConv-ReLU layers, followed by Conv-Sigmoid (Table 2).

     The architecture and hyper-parameters are listed in the corresponding tables. If not specified, use the default values in Pytorch. Train the model by minimizing $-\text{ELBO} = -\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] + D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}))$. You can simply use the Mean Square Error for the reconstruction loss (it is the exact NLL (negative log-likelihood) up to a constant when $\mathbf{x}$ has a Gaussian distribution). You need to do the following:

     - Implement the VAE2 based on the starter code. We will auto-grade your code by evaluating the trained model based on the test set reconstruction loss.
     - Try four different latent dimension $N_z = 2, 5, 10, 20$. Plot sampled images and briefly discuss the effect of the choice of N_z in your report.

Layer	In Channel	Out Channel	kernel size	padding	stride
Conv1-ReLU	1	8	3	1	
MaxPool			2		2
Conv2-ReLU	8	8	3	1	
MaxPool			2		2
Conv3-ReLU	8	8	3	1	
MaxPool			2		2

Table 1: Encoder Conv-ReLU-MaxPool Layers

	In Channel	Out Channel	kernel size	stride	output padding	padding
DeConv1-ReLU	8	8	2	2	1	
DeConv2-ReLU	8	8	2	2		
DeConv3-ReLU	8	8	2	2		
Conv-Sigmoid	8	1	3			1

Table 2: Decoder DeConv-ReLU Layers

Training. You can train these three models on your local machine or a resource like Colab (use of GPU is not necessary). We have fixed most hyper-parameters like batch size, learning rate, etc., except the ones we let you investigate. You need to save the checkpoints (we have provided required code) and upload them along with your Jupyter Notebook.

Visualization We have provided the visualization code in the starter Jupyter Notebook. Please show samples in an 8x8 plot whenever you need to visualize.

Extra Credit Problem.

(EC-1) (10 points) Noisy auto-encoders work with noisy versions of the input by adding noise to the input, which results in convolution of the input and noise distributions. In many cases, the input distribution has a bounded or even discrete support.

(a) (5 points) Consider input X with the following probability density function: $f_X(x) = \frac{1}{2}$ if $x \in [-1, 1]$ and 0 otherwise. Assume the noise E has the standard normal distribution $N(0, 1)$. What is the distribution of $X + E$? Provide precise details and associated derivations. It is okay to present the final answer in terms of standard functions which may not have a closed form.

(b) (5 points) Let the input X have the uniform distribution on $\{-1, +1\}^d$ and the noise E follow the isotropic multivariate Gaussian distribution $N(0, \mathbb{I}_{d\times d})$. What is a non-trivial lower bound of the density function of $X + E$ at the origin? Provide details on how you arrive at the lower bound.