MIT CSAIL

6.s978 Deep Generative Models

Fall 2024

Problem Set 3

Posted: Tuesday, Oct. 08, 2024 **Due:** Tuesday 11:59 pm, Oct. 22, 2024

We provide a Python notebook with the code to be completed. You can run it locally or on Google Colab. To use Colab, upload it to Google Drive and double-click the notebook (or right-click and select Open with Google Colaboratory), which will allow you to complete the problems without setting up your own environment.

Submission Instructions: Please submit two files on Canvas: Submit (1) Your report named `kerberos.pdf` which should include your answers to all required questions with images/plots showing your results as well as the code you wrote (e.g., using screenshots); (2) The provided Python notebook with the relevant source code and with all the cells run.

Attention: If you fail to include your code in your report in your submission, we will not be able to give you credit for your code, so please do not forget.

Late Submission Policy: If your Problem Set is submitted within 7 days of the original deadline, you will receive partial credit. Such submissions will be penalized by a multiplicative coefficient that linearly decreases from 1 to 0.5, step-wise on each day's 11:59pm cutoff.

In this problem set, you will be experimenting with GANs using PyTorch. To reduce training time, we recommend using GPU acceleration. Colab comes with free GPU support. On Colab, select GPU as your runtime type as follows:
Runtime → **Change runtime type** → **Hardware accelerator** → **GPU** → **Save**.

If you exceed your GPU usage limit on Colab, don't fret. You can also complete your problem set using a regular CPU in a reasonable amount of time.

Problem 0 *Notebook Submission, .ipynb version* (1 point, required)

Problem 1 *Unconditional GAN on MNIST* (20 points)

In this problem, you will implement an Unconditional Generative Adversarial Network (GAN) on the MNIST dataset. The goal is to train a GAN that generates images of handwritten digits from random noise vectors.

> Implement the network architecture for the generator and discriminator in the provided Python notebook. Both will consist of fully connected layers, using

Leaky ReLU as the activation function. No batch normalization will be used. The generator uses 128, 256, and 512 units in the linear layers. The discriminator uses 512, 256, and 128 units in the linear layers. The discriminator will employ dropout for regularization, while the generator will not. Plot the loss curves and the generated images after training. You may need to adjust hyperparameters such as the learning rate, number of epochs, and batch size to achieve good results.

Problem 2 *Ablation Study of Unconditional GAN on MNIST* (15 points)

You will run multiple experiments, systematically varying the architecture and training strategies. Specifically, you will:

1. Double the size of the channels in generator.
2. Increase the capacity of the discriminator by increasing the number of layers.
3. Modify the training loop to update the discriminator 10 times for every 1 update of the generator.

For each experiment, train the GAN on the MNIST dataset, plot the loss curves, and evaluate the quality of the generated images. Compare the results of each experiment to the baseline GAN trained in Problem 1.

Problem 3 *Conditional GAN on MNIST* (15 points)

In this problem, you will implement a Conditional Generative Adversarial Network (CGAN) on the MNIST dataset. The CGAN allows for generating images conditioned on specific class labels, enabling the model to produce digits corresponding to the input label. You will modify both the generator and discriminator architectures to incorporate label information.

Implement a Conditional GAN where both the generator and discriminator use embedding layers for label information. For the generator, concatenate the embedded label with the noise vector input. For the discriminator, concatenate the embedded label with the input image. Train the CGAN on the MNIST dataset. Plot the loss curves and the generated images after training.

Problem 4 *Theoretical Derivations related to GAN* (20 points)

In this problem, we will focus on two important results from the theory of GANs as introduced in the lecture. Your task is to provide a detailed derivation for each of these results.

Consider a GAN model where G is the generator and D is the discriminator. Let $p_{\text{data}}(x)$ represent the probability distribution of the real data, and $p_g(x)$ represent the distribution generated by the generator G. D and G play the following two-player minimax game with value function $V(G, D)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]. \tag{1}$$

For a fixed generator G, the optimal discriminator D is given by:

$$D_G^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}$$

(a) Provide a detailed derivation for this optimal discriminator formula. You may start by considering the form of the GAN objective function and use calculus to derive the optimality condition for D.

The global minimum of the virtual training criterion $C(G) = \max_D V(D, G)$ is achieved if and only if $p_g = p_{\text{data}}$. At this point, $C(G)$ achieves the value $-\log 4$.

(b) Provide a detailed derivation of this result. Show that the global minimum of the criterion occurs when $p_g = p_{\text{data}}$, and calculate the corresponding value of $C(G)$.

Problem 5 *Kantorovich-Rubinstein Duality in Discrete Optimal Transport* (20 points)

In the lecture, we introduced the Wasserstein Generative Adversarial Network (W-GAN), which optimizes for the Wasserstein Distance, commonly known as the Earth Mover's Distance (EMD). The Wasserstein Distance has deep mathematical foundations rooted in Optimal Transport (OT) theory, which is an active area of research. Despite its complex mathematical underpinnings, Optimal Transport has practical applications in various fields, including machine learning, image processing, geometry processing, and economics. In this problem, we will explore a specific case of Optimal Transport in the discrete setting and demonstrate the famous Kantorovich-Rubinstein duality.

Consider two finite sets $X = \{1, 2, \ldots, k_1\}$ and $Y = \{1, 2, \ldots, k_2\}$, and two discrete distributions $v \in \mathbb{R}^{k_1}$ and $w \in \mathbb{R}^{k_2}$, such that:

$$\sum_{i=1}^{k_1} v_i = \sum_{j=1}^{k_2} w_j = 1.$$

Let $C = [c_{ij}]$ be a cost matrix, where c_{ij} represents the cost of transporting mass from element $i \in X$ to element $j \in Y$.

The Optimal Transport (OT) problem can be written as the following linear program:

$$\begin{aligned}
\mathrm{OT}(v, w; C) = \min_{T \in \mathbb{R}^{k_1 \times k_2}} & \sum_{i,j} T_{ij} c_{ij} \\
\text{s.t.} \quad & T_{ij} \geq 0, \\
& \sum_j T_{ij} = v_i, \forall i \in \{1, \ldots, k_1\}, \\
& \sum_i T_{ij} = w_j, \forall j \in \{1, \ldots, k_2\}.
\end{aligned}$$

The Kantorovich-Rubinstein duality theorem provides a way to express the Optimal Transport problem in its dual form, transforming it into a simpler optimization problem over dual variables.

$$
\begin{aligned}
\mathrm{OT}(v, w; C) = \max_{\phi,\psi} & \sum_{i=1}^{k_1} v_i \phi_i + \sum_{j=1}^{k_2} w_j \psi_j \\
\text{s.t.} \quad & \phi_i + \psi_j \leq c_{ij}, \quad \forall i \in \{1, \ldots, k_1\}, j \in \{1, \ldots, k_2\}.
\end{aligned}
$$

Here, $\phi \in \mathbb{R}^{k_1}$ and $\psi \in \mathbb{R}^{k_2}$ are dual potential functions, and the dual problem seeks to maximize their sum subject to the coupling constraint.

(c) Derive the dual form of the discrete Optimal Transport problem above. To help you with the derivation, recall that the dual of a convex optimization problem can often be derived by introducing Lagrange multipliers for the equality constraints in the primal problem and maximizing the resulting Lagrangian with respect to the dual variables. Check the Wikipedia link of KKT condition for more details: `https://en.wikipedia.org/wiki/Karush%E2%80%93Kuhn%E2%80%93Tucker_conditions`