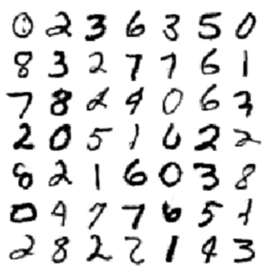
CSDS 600: Deep Generative Models

Homework 2
Instructor: Yu Yin
Due Date: 11:59 pm Oct. 15
Submission: Canvas

1. **Normalizing Flows (20/100)**
   a. Describe the training objective for flows.
   b. Describe the sampling process of normalizing flows.
   c. What properties does the transformation function f_θ must satisfy for practical flow models?

2. **Wasserstein Distance (35/100)**
   In many cases, the GAN algorithm can be thought of as minimizing a divergence between a data distribution p_{data} and the model distribution p_g. In this problem, we will explore an issue with various divergences (e.g., Jensen-Shannon divergence and KL divergence) and one potential way to fix it.
   a. Let $p_{data} \sim \mathcal{N}(\theta_0, \epsilon^2)$ and $p_g \sim \mathcal{N}(x|\theta, \epsilon^2)$ be normal distributions with standard deviation ϵ centered at $\theta_0 \in \mathbb{R}$ and $\theta \in \mathbb{R}$, respectively. Show that
   $$D_{KL}(p_g || p_{data}) = \frac{(\theta - \theta_0)^2}{2\epsilon^2}.$$
   b. Suppose p_{data} and p_g both place probability mass in only a very small part of the domain; that is, consider the limit $\epsilon \to 0$. What happens to $D_{KL}(p_g || p_{data})$ and its derivative with respect to θ, assuming that $\theta \neq \theta_0$?
   c. Would this pose an issue for a GAN trained with the loss function defined below? Why?
   $$L_G(\theta; \phi) = \mathbb{E}_{\boldsymbol{x} \sim p_\theta(\boldsymbol{x})}[\log(1 - D_\phi(\boldsymbol{x}))] - \mathbb{E}_{\boldsymbol{x} \sim p_\theta(\boldsymbol{x})}[\log D_\phi(\boldsymbol{x})]$$
   d. Under the same circumstances as (b), compare KL divergence, JS divergence and Wasserstein Distance.

3. **Wasserstein GAN for MNIST (45/100)**
   In this question, you will train a Wasserstein GAN on the MNIST dataset, which comprises 28 × 28 grayscale images.
   a. Implement the optimization process for the Wasserstein GAN. Consider making multiple updates (e.g., 2) to the discriminator for each update to the generator.
   b. Train your model for 40 epochs (we recommend using a learning rate of lr = 3e-4). After training, the generated samples should resemble the results shown below.



   Request deliverables:
   a. Provide 100 samples from your trained WGAN.
   b. Report the Inception Score (https://github.com/openai/improved-gan/tree/master/inception_score).
   c. Include Interpolations between two random latent variables.