ECE 689 Spring 2024 - HW1

The Jupyter Notebook template for this homework is available on Sakai. Please upload your Jupyter Notebook file (.ipynb file) with all outputs and necessary derivations exported as a PDF or HTML to Sakai. All necessary coding documentation can be found in `https://pytorch.org/docs/stable/index.html`. **We highly encourage you to submit your derivation in LaTeX. If you choose to submit a hand-written derivation, please make sure your derivation is legible. If you have hand-written solutions, please scan/take a photo and insert them into their designated space.** You can follow this tutorial to insert images to .ipynb files: `https://www.geeksforgeeks.org/insert-image-in-a-jupyter-notebook/`

1 Problem 1: Normalizing Flow (20 pts)

Select a coupling Normalizing Flow and an autoregressive Normalizing Flow (NF). Apply the NF models to MNIST dataset (`https://git-disl.github.io/GTDLBench/datasets/mnist_datasets/`). To reduce the computational complexity, you can resize the MNIST images to 7×7 pixels. Hint: Examples are given at `https://github.com/VincentStimper/normalizing-flows`

2 Problem 2: WaveNet for Image Modeling (30 points)

Choose a WaveNet-typed model (e.g., WaveNet, ParallelWaveNet). Train the model for image modeling on the MNIST dataset. (`https://git-disl.github.io/GTDLBench/datasets/mnist_datasets/`). To reduce the computational complexity, you can resize the MNIST images to 7×7 pixels. After training, please provide synthetic images by sampling the learned distribution. Some references can be found below:

- `https://medium.com/@evinpinar/wavenet-implementation-and-experiments-2d2ee57105d5`
- `https://github.com/kan-bayashi/ParallelWaveGAN?tab=readme-ov-file`
- `https://github.com/Zeta36/tensorflow-image-wavenet`

3 Problem 3: Markov chain Monte Carlo (20 points)

Apply the simple random-walk Metropolis-Hasting and Metropolis Adjusted Langevin Algorithm (MALA) to the following target unnormalized distribution:

$$exp\left(-\sum_i x_i^4 - 0.5\sum_{i\neq j} x_i^2 x_j^2\right)$$

- Provide the histogram of the samples and the trace plot of the samples to see how the chain moved around.
- Compare the performance of these approaches.

4 Problem 4: Energy-based Model (30 points)

Design a simple energy-based model (e.g., exp(-NN)). Applying any of the Monte Carlo sampling methods, train this model on the USPS dataset (`https://git-disl.github.io/GTDLBench/datasets/usps_dataset/`).