ECE 689 Spring 2025 - HW1

The Jupyter Notebook template for this homework is available on Canvas. Please upload your Jupyter Notebook file (.ipynb file) with all outputs and necessary derivations exported as a PDF or HTML to Canvas. 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 (30 pts)

Implement two types of Normalizing Flow (NF) models:

1. Coupling Normalizing Flow
2. Autoregressive Normalizing Flow

You can select or design your models. Next, these NF models will be applied to the MNIST dataset. The dataset can be found via `https://git-disl.github.io/GTDLBench/datasets/mnist_datasets/`. To reduce the computational complexity, you can downsampling the MNIST images (e.g., resize images to 7×7 pixels).

After training, sample synthetic images from the learned distributions of both models. Particularly, plot a grid of 8 by 8 images generated by these NF models. Additionally, report the log-likelihood on the test data for both models.

Hint: Examples are given in `https://github.com/VincentStimper/normalizing-flows`.

2 Problem 2: WaveNet for Image Modeling (40 points)

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

Some references and examples 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: Energy-based Model (30 points)

Design a simple energy-based model for image modeling. The model is defined by an energy function $E(x;\theta) = f_\theta(x)$, where f is a 2-layer convolutional network, and θ are the parameters. The probability of an image x is expressed as:

$$P(x;\theta) = \frac{\exp(-E(x;\theta))}{\int \exp(-E(x;\theta))dx}.$$

Applying two Monte Carlo sampling methods: (1) Gibbs sampling, (2) Langevin Dynamics, train this model on the USPS dataset. Compare the results of these two sampling methods. The dataset can be found via `https://git-disl.github.io/GTDLBench/datasets/usps_dataset/`.