CSE 599i (Generative Models) Homework 3

Analysis

Problem 1. (Estimating the MLE) We initially abandoned maximum likelihood estimation for learning a pushforward distribution because computing the likelihood was too difficult. We can implicitly compute a KL-divergence (and thus the MLE) using an f-GAN. Show that

$$D(p \parallel q) = 1 + \sup_{d:\mathcal{X}\to\mathbb{R}} \left[\mathop{\mathbb{E}}_{x\sim p} \log d(x) - \mathop{\mathbb{E}}_{x\sim q} d(x) \right].$$

Hint: Let $f(x) = x \log x$, $T(x) = 1 + \log d(x)$, and apply the proposition from Lecture 10.

Problem 2. (Estimating the Reverse-KL) Show that

$$D(q \parallel p) = 1 + \sup_{d:\mathcal{X}\to\mathbb{R}} \left[\mathop{\mathbb{E}}_{x\sim p} d(x) + \mathop{\mathbb{E}}_{x\sim q} \log(-d(x)) \right].$$

Hint: let $f(x) = -\log x$ and apply the proposition from Lecture 10.

Problem 3. (Wasserstein Distance is Smooth) Let $p : \mathbb{R}^d \to \mathbb{R}$ be a probability density on $\mathbb{R}^d$, and let $p_\sigma : \mathbb{R}^d \to \mathbb{R}$ denote the density of $y = x + \varepsilon$, where $x \sim p$ and $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$. Recall that the Wasserstein distance between two probability densities is defined by

$$W_1(p, q) = \inf_{\pi\in\Pi(p,q)} \mathop{\mathbb{E}}_{(x,y)\sim\pi} \left[\|x - y\|_2 \right].$$

Prove that $W_1(p, p_\sigma) \leq \sigma\sqrt{d}$.

Problem 4. (Wasserstein Distance is a Metric) Recall that the discrete Wasserstein distance on a space with n elements and cost metric c is defined by

$$W_1(p, q) = \min_{\pi\in\Pi(p,q)} \langle c, \pi \rangle.$$

Verify the triangle inequality for the discrete Wasserstein distance:

$$W_1(p, q) \leq W_1(p, r) + W_1(r, q).$$

Hint: let $\pi_p = \mathop{\arg\min}_{\pi\in\Pi(p,r)} \langle c, \pi \rangle$, $\pi_q = \mathop{\arg\min}_{\pi\in\Pi(r,q)} \langle c, \pi \rangle$, and define $\kappa = \pi_p \operatorname{diag}(1/r) \pi_q$. Show that

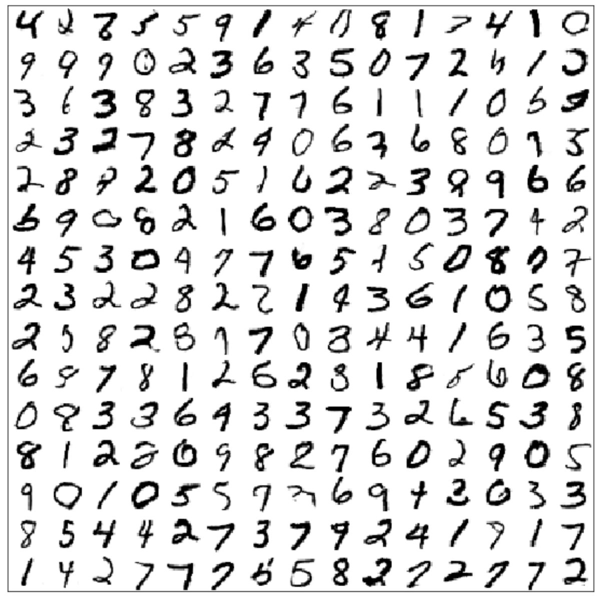
$$W_1(p, q) \leq \langle \kappa, c \rangle \leq W_1(p, r) + W_1(r, q).$$

Hint: Because c is a cost metric, it satisfies the triangle inequality $c(x, y) \leq c(x, z) + c(z, y)$.

Implementation

See the github repository for framework code. Turn in your version of `models.py`, your `mnist.ipynb` and `cifar10.ipynb` notebooks, and plots of your results. Hacks to notice (but shouldn't affect your work): we don't use BatchNorm in the discriminator (it doesn't interact well with Lipschitz regularization) and we set $\beta_1 = 0$ in the Adam optimizer (the momentum seems to destabilize training).

Problem 5. (Wasserstein GAN for MNIST) In this problem you will train a Wasserstein GAN on the MNIST dataset, using the training framework found in `mnist.ipynb`. This dataset consists of 28×28 grayscale images.



Part a. Implement the Wasserstein GAN optimization updates in `mnist.ipynb`. People usually make multiple updates to the discriminator for each update to the generator, but I find alternating between minimization and maximization steps works fine. Enforce the Lipschitz constraint using gradient penalty (implemented in `models.py`) with a Lagrange multiplier $\lambda = 10$.

Part b. Train your model for 40 epochs. After training, samples from your model should look like the results to the right. How good are these results? That's hard to quantify.

Problem 6. (Wasserstein GAN for CIFAR-10) In this problem you will train a Wasserstein GAN on the CIFAR-10 dataset, using the training framework found in `cifar10.ipynb`. This dataset consists of $3 \times 32 \times 32$ rgb color images.

Part a. Make minimal updates to the `Generator` and `Discriminator` classes in `models.py` to accommodate $3 \times 32 \times 32$ CIFAR-10 data. To account for higher complexity of CIFAR-10 data, increase the generator's residual blocks from 3 to 9.

Part b. Using the same Wasserstein GAN optimization that you implemented in Problem 5, train your model for at least 100 epochs, cutting the learning rate to 3e-5 after 80 epochs. You should get an Inception score around 7.9. If you have the time: train for 300 epochs, cutting the learning rate to 3e-5 after the first 200 epochs. You'll get results like the ones to the right with Inception score around 8.2.