CSE 599i (Generative Models) Homework 1

Analysis

Problem 1. (Location-Scale Transformation) Let $g : \mathbb{R} \to \mathbb{R}$ be defined by $z \mapsto \mu + \sigma z$ where $\mu, \sigma \in \mathbb{R}$ and $\sigma > 0$. Prove that if $z \sim \mathcal{N}(0,1)$ then $g(z) \sim \mathcal{N}(\mu, \sigma^2)$.

Problem 2. (Gumbel-Argmax Sampling) Let p be a discrete distribution on the finite space $\mathcal{X}$ with n elements. Define $g : (0,1)^n \to \mathcal{X}$ by

$$z \mapsto \arg\max_{x \in \mathcal{X}} \left(\log p(x) - \log\log \frac{1}{z_x} \right).$$

Prove that if $z \sim \text{Uniform}(0,1)^n$ i.i.d. then $g(z) \sim p$. Hint: $-\log\log\frac{1}{z_x} \sim \text{Gumbel}(0,1)$.

Problem 3. (Discrete Maximum Likelihood Estimation) Let p be a discrete distribution on a finite space $\mathcal{X}$ and define

$$\pi^* = \arg\max_{\pi \in \Delta^{|\mathcal{X}|-1}} \mathop{\mathbb{E}}_{x \sim p} \left[\log \pi_x\right].$$

Show that $\pi^*_x = p(x)$ for all $x \in \mathcal{X}$. Hint: consider using the information inequality $D(p \parallel q) \geq 0$.

Problem 4. (The M-step of EM for GMMs). Let $p_\theta(x, z)$ be a K-Gaussian mixture model with parameters $\theta = (\pi, \mu)$ defined by the generative process

1. $z \sim \text{Categorical}_\pi(K)$,
2. $x \sim \mathcal{N}(\mu_z, I)$.

Consider the ELBO optimization problem:

$$\pi', \mu' = \arg\max_{\pi,\mu} \mathop{\mathbb{E}}_{x\sim p} \left[\mathop{\mathbb{E}}_{z \sim q(\cdot|x)} \big[\log p_\theta(x|z)\big] - D(q(z|x) \parallel p_\theta(z)) \right].$$

Prove that, if we approximate the expectation $\mathop{\mathbb{E}}_{x\sim p}$ with samples $x_1, \ldots, x_n \sim p$, then

$$\pi'_k = \frac{1}{n}\sum_{i=1}^{n} q(z_k|x_i), \qquad \mu'_k = \frac{1}{n\pi'_k}\sum_{i=1}^{n} q(z_k|x_i)x_i.$$

Problem 5. (Autoregressive Mean Estimation) Recall the AR(1) model defined in Lecture 3, where a sequence $\mathbf{x} \in \mathbb{R}^{T\times 1}$ is governed by the following linear dynamics ($|\rho| < 1$):

$$\begin{aligned} x_t &= \rho(x_{t-1} - \mu) + \mu + \varepsilon_t, \\ x_1 &= \mu + \varepsilon_1. \end{aligned}$$

Assume that $\varepsilon_1 \sim \mathcal{N}(0, \frac{\sigma^2}{1-\rho^2})$ and that $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$ for $t > 1$. Verify the following claims in the lecture notes: $\mathbb{E}[x_t] = \mu$, $\text{Var}[x_t] = \frac{\sigma^2}{1-\rho^2}$, and $\gamma(k) = \frac{\rho^k\sigma^2}{1-\rho^2}$.

Problem 6. (Time Series Estimation) Construct an example of a time series for which the sample mean is an *inconsistent* estimator of the process mean.

Implementation

Problem 7. (Gaussian Mixture Models) In this exercise, you will fit the means $\theta = \{\mu_1, \mu_2, \mu_3\}$ of a GMM (assume that $\pi_k = 1/3$ and Σ_k is the identity), using the data generated in Part a. You will try three different optimization techniques to approximate the maximum likelihood estimator

$$\hat{\theta}_{\text{mle}} = \arg\max_{\theta} \sum_{i=1}^{n} \log p_\theta(x_i).$$

You should find that each algorithm produces comparable numerical results, with a final negative log-likelihood of around 3.9 nats. You can verify that each algorithm has converged well by plotting the final estimates of the means superimposed on the data. Hint: use PyTorch's automatic differentiation to compute gradients, rather than trying to take them by hand.

Part a. Sample 1,000 points from each of three Gaussian distributions in $\mathbb{R}^2$ with means $\mu_0 = (-2, 0)$, $\mu_1 = (1, 3)$, and $\mu_2 = (2, -2)$ and identity covariance ($n = 3,000$ total points). Plot the points in a 2-d scatter plot.

Part b. Implement gradient ascent to approximate the MLE of θ . Plot the log-likelihood of the data using the estimators $p_{\theta^{(k)}}$ as a function of $k = \{0, \dots, 10\}$, where $\theta^{(k)}$ are the values of the parameters after the k'th iteration of gradient descent.

Part c. Implement stochastic gradient ascent (SGD) to approximate the MLE of θ. Plot the log-likelihood of the data using the estimators $p_{\theta^{(k)}}$ as a function of $k = \{0, \dots, 5000\}$, where $\theta^{(k)}$ are the values of the parameters after the k'th iteration of SGD.

Part d. Implement expectation-maximization (EM) to approximate the MLE of θ, using the result of Problem 4. Plot the log-likelihood of the data using the estimators $p_{\theta^{(k)}}$ as a function of $k = \{0, \dots, 10\}$, where $\theta^{(k)}$ are the values of the parameters after the k'th iteration of EM.

Problem 8. (Wikitext-2) See the github repository for framework code. I encourage you to read the code in the context of the discussion we've had in class (there's not that much code!). There should be no surprising hacks in this codebase; if something does seem surprising, please bring it up on the class discussion board and we can talk about it.

Part a. Implement the forward pass of the the `TransformerBlock` found in **transformer.py**.

Part b. Train your transformer for 10 epochs on Wikitext-2 using 2 layers and 2 attention heads, and no dropout. Report the test set log-likelihood of the model.

Part c. Train your transformer for 10 epochs on Wikitext-2 using 16 layers and 10 attention heads, and no dropout. Report the test set log-likelihood of the model.

Part d. Train your transformer for 80 epochs on Wikitext-2 using 16 layers, 10 attention heads, 20% dropout, and 60% input/output dropout. Report the test set log-likelihood of the model.