CS480/680: Introduction to Machine Learning

Homework 4

Due: 11:59 pm, July 24, 2024, submit on LEARN.

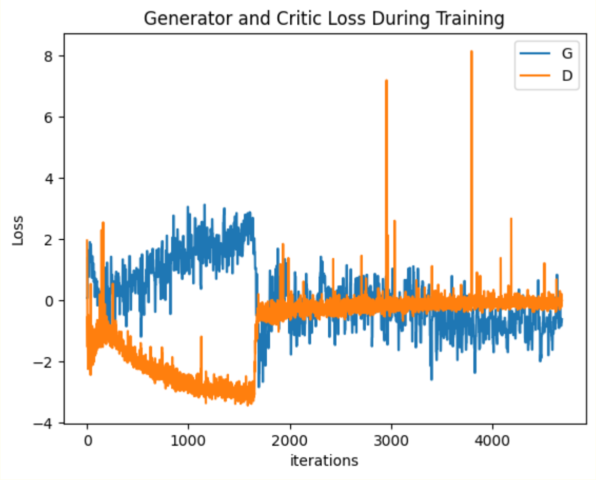
Prajwal Thakur
21052875

Submit your writeup in pdf and all source code in a zip file (with proper documentation). Write a script for each programming exercise so that the TA can easily run and verify your results. Make sure your code runs!
[Text in square brackets are hints that can be ignored.]

Exercise 1: Generative Adversarial Networks (10 pts)

Please follow the instructions of this `ipynb` file.

1. (2+6 pts) Complete the missing coding parts in the provided `ipynb` file.
2. (1 pt) Visualization of the loss:



3. (1 pt) Visualization of the final generated images:

Exercise 2: Quantile and push-forward (8 pts)

In this exercise we compute and simulate the push-forward map T that transforms a reference density r into a target density p. Recall that the quantile function of a (univariate) random variable X is defined as the inverse of its cumulative distribution function (cdf) F:

$$F(x) = \Pr(\mathsf{X} \le x), \qquad Q(u) = F^{-1}(u), \quad u \in (0, 1). \tag{1}$$

We assume F is continuous and strictly increasing so that $Q^{-1} = F$. A nice property of the quantile function, relevant to sampling, is that if $\mathsf{U} \sim \text{Uniform}(0, 1)$, then $Q(\mathsf{U}) \sim F$.

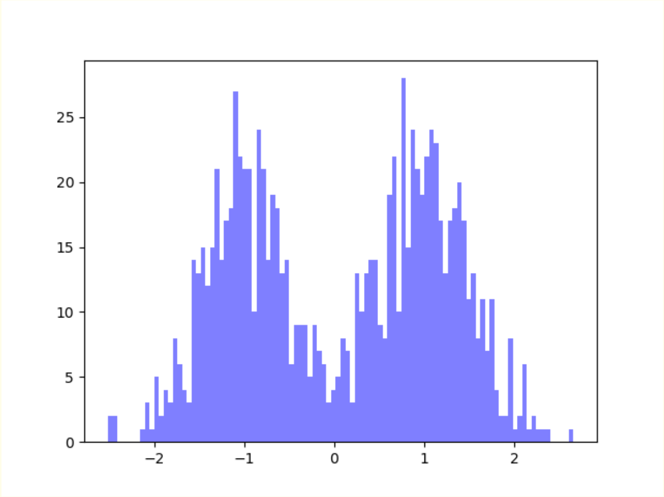
In the following, do not confuse cdf (signaled by uppercase letters) with pdf (i.e., density, signaled by lowercase letters).

1. (1 pt) Consider the Gaussian mixture model (GMM) with density $p(x) = \frac{\lambda}{\sigma_1}\varphi\left(\frac{x-\mu_1}{\sigma_1}\right) + \frac{1-\lambda}{\sigma_2}\varphi\left(\frac{x-\mu_2}{\sigma_2}\right)$, where φ is the *density* of the standard normal distribution (mean 0 and variance 1). Implement the following to create a dataset of $n = 1000$ samples from the GMM p:

   - Sample $\mathsf{U}_i \sim \text{Uniform}(0, 1)$.
   - If $\mathsf{U}_i < \lambda$, sample $\mathsf{X}_i \sim \mathcal{N}(\mu_1, \sigma_1^2)$; otherwise sample $\mathsf{X}_i \sim \mathcal{N}(\mu_2, \sigma_2^2)$.

   Plot the histogram of the generated X_i (with $b = 50$ bins) and submit your script as
   `X = GMMsample(gmm, n=1000, b=50), gmm.lambda=0.5, gmm.mu=[1,-1], gmm.sigma=[0.5,0.5]`

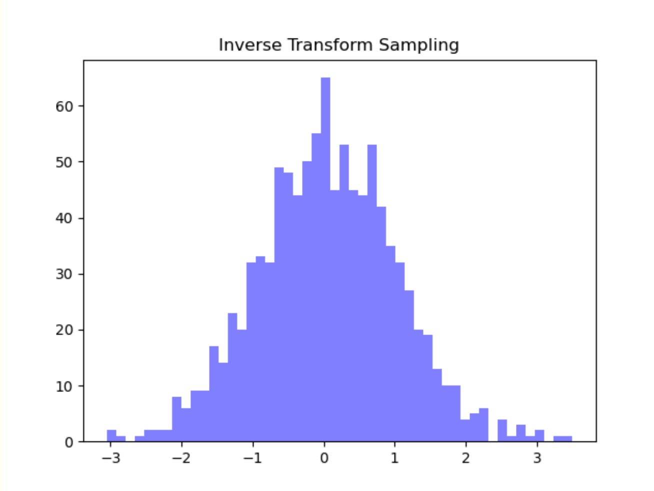
   [See here or here for how to plot a histogram in `matplotlib` or `pandas` (or `numpy` if you insist).]

   Ans: 

2. (2 pts) Compute $\mathsf{U}_i = \Phi^{-1}\big(F(\mathsf{X}_i)\big)$, where F is the *cdf* of the GMM in Ex 2.1 and Φ is the *cdf* of standard normal. Plot the histogram of the generated U_i (with b bins). From your inspection, what distribution should U_i follow (approximately)? Submit your script as `GMMinv(X, gmm, b=50)`.

   [This page may be helpful.]

   Ans: From Observation it seems that U_i should follow a standard uni-variate Normal Distribution with mean=0, and std-deviation=1. Function F will map the random variable map it to (0,1) and now passing by inverse cdf of standard normal distribution to get the sample back from standard normal distribution.



3. (2 pts) Let $\mathsf{Z} \sim \mathcal{N}(0, 1)$. We now compute the push-forward map T so that $T(\mathsf{Z}) = \mathsf{X} \sim p$ (the GMM in Ex 2.1). We use the formula:

$$T(z) = Q(\Phi(z)), \tag{2}$$

where Φ is the *cdf* of the standard normal distribution and $Q = F^{-1}$ is the quantile function of X, namely the GMM p in Ex 2.1. Implement the following binary search Algorithm 1 to numerically compute T. Plot the function T with input $z \in [-5, 5]$ (increment 0.1). Submit your main script as `BinarySearch(F, u, lb=-100, ub=100, maxiter=100, tol=1e-5)`, where `F` is a function. You may need to write another script to compute and plot T (based on `BinarySearch`).

Algorithm 1: Binary search for solving a monotonic nonlinear equation $F(x) = u$.

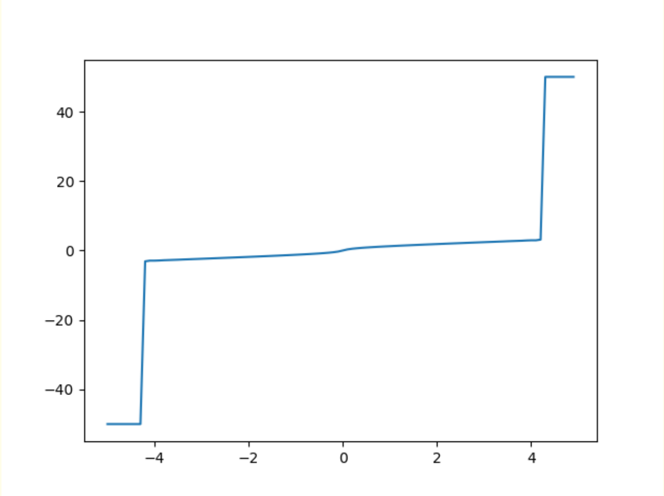
Input: $u \in (0, 1)$, $\texttt{lb} < 0 < \texttt{ub}$, `maxiter`, `tol`

Output: x such that $|F(x) - u| \leq \texttt{tol}$

while $F(\texttt{lb}) > u$ **do** // lower bound too large
- $\texttt{ub} \leftarrow \texttt{lb}$
- $\texttt{lb} \leftarrow 2 * \texttt{lb}$

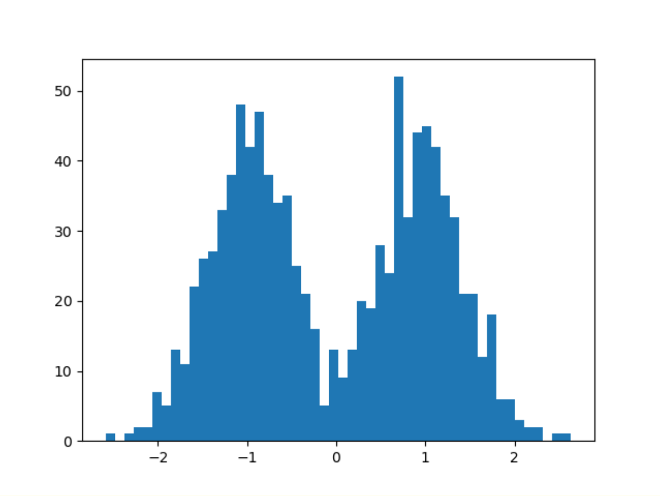
while $F(\texttt{ub}) < u$ **do** // upper bound too small
- $\texttt{lb} \leftarrow \texttt{ub}$
- $\texttt{ub} \leftarrow 2 * \texttt{ub}$

for $i = 1, \ldots, \texttt{maxiter}$ **do**
- $x \leftarrow \frac{\texttt{lb}+\texttt{ub}}{2}$ // try middle point
- $t \leftarrow F(x)$
- **if** $t > u$ **then**
  - $\texttt{ub} \leftarrow x$
- **else**
  - $\texttt{lb} \leftarrow x$
- **if** $|t - u| \leq \texttt{tol}$ **then**
  - **break**

Ans: 

4. (2 pts) Sample (independently) $\mathsf{Z}_i \sim \mathcal{N}(0,1), i = 1, \ldots, n = 1000$ and let $\tilde{\mathsf{X}}_i = T(\mathsf{Z}_i)$, where T is computed by your `BinarySearch`. Plot the histogram of the generated $\tilde{\mathsf{X}}_i$ (with b bins) and submit your script as `PushForward(Z, gmm)`. Is the histogram similar to the one in Ex 2.1?

   Ans: yes , it same as 2.1 since here too we are doing invverse transform sampling through Algo.1 $T(z) = Q(\Phi(z)) \equiv F^{-}(\Phi(z))$ where $\Phi(z)$ will give the value b/w (0,1) then F^{-1} being the inverse cdf of GMM will

   give the random sample drawn from GMM .

5. (1 pt) Now let us compute $\tilde{\mathsf{U}}_i = \Phi^{-1}(F(\tilde{\mathsf{X}}_i))$ as in Ex 2.2, with $\tilde{\mathsf{X}}_i$'s being generated in Ex 2.4. Plot the histogram of the resulting $\tilde{\mathsf{U}}_i$ (with b bins). From your inspection what distribution should $\tilde{\mathsf{U}}_i$ follow (approximately)? [No need to submit any script, as you can recycle `GMMinv`.]

   Ans: $\tilde{\mathsf{X}}_i$ is random variables we pass it to the cdf of GMM to get the values b/w (0,1) then pass it to the inverse cdf function of standard normal distribution to get the random samples from standard normal distribution .