Problem Set 3

Important:

- Write your name as well as your NU ID on your assignment. Please number your problems.
- Submit both results and your code.
- Give complete answers. Do not just give the final answer; instead show steps you went through to get there and explain what you are doing. Do not leave out critical intermediate steps.
- This assignment must be submitted electronically through Gradescope by November 7th 2025 (Friday) by 11:59 PM.

1 Deeper Dive Into Attention

2 KL Divergence and Maximum Likelihood

In Machine Learning, we often have access to some observed data, say $\mathcal{D}$, as a finite set of samples from an underlying distribution, say p_{data}. We are interested in parametric approximations to the data distribution, which summarize all the information about the dataset $\mathcal{D}$ in a finite set of parameters. We can think of the task of learning a model as picking the parameters within a family of model distributions that minimizes some notion of distance between the model distribution and the data distribution.

For instance, we might be given access to a dataset of dog images and our goal is to learn the parameters of a model θ within a model family $\mathcal{M}$ such that the model distribution p_θ is close to the data distribution over dogs p_{data}. Mathematically, we can specify our goal as the following optimization problem,

$$\min_{\theta \in \mathcal{M}} \; d(p_{data}, p_\theta),$$

where p_{data} is accessed via the dataset $\mathcal{D}$ and $d(\cdot, \cdot)$ is a notion of distance between probability distributions. One common distance function that is adopted is the KL divergence which we explore further in this problem. The final goal of this problem is to show that Maximum Likelihood Estimation, which is a commonly used optimization objective, turns out to be equivalent to minimizing the KL divergence between the empirical distribution p_{data} and the model distribution p_θ.

The KL divergence two distributions $p(X)$ and $q(X)$ over a space $\mathcal{X}$ is defined by,

$$D_{KL}(p, q) = \sum_{x \in \mathcal{X}} p(x) \log \Big[\frac{p(x)}{q(x)}\Big].$$

In class, we derived the expression of the KL divergence by looking at the entropy, measure of uncertainty, and cross-entropy of distributions.

1. Show that for any distributions $p(x)$ and $q(x)$, we always have $D_{KL}(p, q) \geq 0$. This is necessary as a distance always needs to be non-negative. You will need to apply Jensen's inequality which states that if f is a convex function, and Z is a random variable, then $E[f(Z)] \geq f(E[Z])$.
Hint: Start by showing that $D_{KL}(p, q) = E_{p(x)}\Big[- \log \big(\frac{q(x)}{p(x)}\big)\Big]$. Also, note that $-log(\cdot)$ is a convex function so that you can use Jensen's inequality.

2. Show that the KL divergence between two distributions $p(X)$ and $q(X)$ is 0 if and only if $p(x) = q(x)$. This is expected as the "distance" between $p(X)$ and $q(X)$ is 0 if and only if these distributions are identical. You will need to prove the following two results to establish the equivalence:

- If $D_{KL}(p, q) = 0$, then $p(x) = q(x)$ for all x. This part will require a special case of Jensen's inequality which states that if f is strictly convex ($f(x) = -\log(x)$ is strictly convex for example), Z is a random variable and $E[f(Z)] = f(E[Z])$, then $Z = E(Z)$, that is, Z is a non-random constant since $E(Z)$ is a non-random constant.
  Hint: Let $Z = \frac{q(x)}{p(x)}$ and $f(x) = -\log(x)$. Can you relate $E_{p(x)}\Big[-\log\big(\frac{q(x)}{p(x)}\big)\Big]$ to $-\log\Big(E_{p(x)}\Big[\frac{q(x)}{p(x)}\Big]\Big)$?
- If $p(x) = q(x)$ for all x, then $D_{KL}(p, q) = 0$.

3. We provide you with a training set $\mathcal{D}_{train} = \{\vec{x}^{(1)}, \vec{x}^{(2)}, ..., \vec{x}^{(n)}\}$ such that $\vec{x}^{(i)} \in \mathbb{R}^k$ for all $i = 1, ..., n$ and $k > 0$. We assume that the probability density function of the data distribution is a uniform distribution over the training set. That is,

$$p_{data}(\vec{x}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(\vec{x}^{(i)} = \vec{x}).$$

We consider a family of models $\mathcal{M}$ with distributions p_θ parametrized by θ. Prove that finding the maximum likelihood estimate for the parameter θ is equivalent to finding p_θ with minimal KL divergence from p_{data}. That is, show that,

$$\arg\min_\theta \; D_{KL}(p_{data}, p_\theta) = \arg\max_\theta \; \sum_{i=1}^{n} \log\Big(p_\theta(\vec{x}^{(i)})\Big). \tag{1}$$

The right hand side of the above equation indicates that we are maximizing the log-likelihood function.

3 Mixture of Gaussians variational autoencoder GMVAE

In this problem, you will be using PyTorch to implement a variant of the variational autoencoder (VAE) and learn a probabilistic model of the MNIST dataset of handwritten digits. We observe a sequence of binary pixels $\vec{x} \in \mathbb{R}^d$ with $x_i \in \{0, 1\}$ for $i = 1, ..., d$, and we assume that we have k latent variables so that $\vec{z} \in \mathbb{R}^k$.

In class, we implemented the model defined by,

$$\begin{aligned}
p(\vec{z}) &= \mathcal{N}(\vec{0}, I),\\
p_\theta(\vec{x}|\vec{z}) &= Bern\big(f_\theta(\vec{z})\big),\\
q_\lambda(\vec{z}|\vec{x}) &= \mathcal{N}\Big(\mu_\lambda(\vec{x}), diag\big(\sigma_\lambda(\vec{x})\big)\Big).
\end{aligned}$$

In this problem, the prior distribution $p(\vec{z})$ will be given by a mixture of Gaussians to obtain more expressivity and better model our data. That is,

$$p(\vec{z}) = \frac{1}{k}\sum_{i=1}^{k} \mathcal{N}(\vec{\mu_i}, diag(\sigma_i^2)),$$

where $i \in \{1, ..., k\}$ denotes the ith cluster index. The distributions of $p(\vec{x}|\vec{z})$ and $q_\lambda(\vec{z}|\vec{x})$ remain identical to the original VAE model.

The ELBO for the GMVAE model, given by

$$E_{q_\lambda(\vec{z}|\vec{x})}\Big[\log(p_\theta(\vec{x}|\vec{z}))\Big] - D_{KL}\Big(q_\lambda(\vec{z}|\vec{x}), p(\vec{z})\Big),$$

remains the same as in the VAE model. The main distinction is that KL divergence term cannot be computed analytically between a Gaussian distribution $q_\lambda(\vec{z}|\vec{x})$ and a mixture of Gaussians $p(\vec{z})$. We can estimate the KL divergence term using Monte Carlo sampling resulting in,

$$D_{KL}\Big(q_\lambda(\vec{z}|\vec{x}), p(\vec{z})\Big) \approx \log\Big(q_\lambda(\vec{z}^{(sample)}|\vec{x})\Big) - \log\Big(p(\vec{z}^{(sample)})\Big),$$

where $\vec{z}^{(sample)} \sim q_\lambda(\vec{z}|\vec{x})$ denotes one sample from the encoder distribution.

1. Implement the ***log_ normal_ mixture*** function in ***utils.py*** and the function ***negative_ elbo_ bound*** in ***gmvae.py***. The function ***log_ mean_ exp*** in ***utils.py*** will be helpful and is implemented in way to ensure that your results are numerically stable.

2. To test your implementation, run ***python run_ gmvae.py*** to train the GMVAE. Once the training is complete, after 20000 iterations, it will output the average negative ELBO, the KL term, and reconstruction loss as evaluated on a selected test subset. Report the three numbers you obtain as part of the write-up. The negative ELBO should be around 97.5.

3. Visualize 200 digits by generating a single image tiled in a grid of 10×20 digits sampled from $p(\vec{x})$.