作业一

1，在进行数值计算的时候我们常听到 Rounding Error, Underflow, Overflow 这些概念。我们在优化模型的时候也常会需要计算 $log(sum(exp(a_{i:n})))$这样形式的式子。如果直接计算，向量 a 里面有大的数值时，exp 会 overflows (Inf); 向量 a 里面的数字都是很小的负数时，log 会 underflow (-Inf);为了克服这些数值计算的问题，请提出一个通用的解决办法，描述思路和数学式子，并证明该计算方法可以得到**精确的结果**；用 Python 实现该函数，并提供一些运行例子（数学算法推导请用该 word 提交；函数代码、运行例子和计算结果请放在 python notebook 随 word 一并提交）。

对应 markdown 和 latex 源码在 jupyter nootbook 中，此处就以图片的形式展示啦。函数代码、实例、结果在 notebook 中

第1题

为了保证不上溢、下溢。对原公式做出如下修改

$$
\begin{aligned}
log \sum_{i=1}^{N} e^{x_i} &= log(e^{x_1} + e^{x_2} + ... + e^{x_N}) \\
&= log(e^a \cdot (e^{x_1 - a} + e^{x_2 - a} + ... + e^{x_N - a})) \\
&= a + \log \sum_{i=1}^{N} e^{x_i - a}
\end{aligned}
$$

其中，a的值取

$$a = \max_{i \in [N]} x_i$$

保证了e的指数最大为0，不会上溢.下溢的结果会被认为是0（上溢会报错，或变成负数）

2. 假设你构建了一个 CNN 模型，里面主要用到 CONV 和 POOL 操作，具体的操作注释如下：

• CONV-K-N 表示 CONV layer 含 N 个 K×K 的 filters，Padding 0, Stride 1

• POOL-K 表示 K × K pooling layer, Stride K, Padding 0

• FC-N 表示有 N neurons 的 fully-connected layer

a) 请计算下面模型每个 layer 的输出维度，参数数目，和 bias 的数目

Layer	Output dimensions	Number of weights	Number of biases
INPUT	128×128×3 (3 是 channel）	0	0
CONV-9-32	120×120×32	3×9×9×32	32
POOL-2	60×60×32	0	0
CONV-5-64	56×56×64	32×5×5×64	64
POOL-2	28×28×64	0	0
CONV-5-64	24×24×64	64×5×5×64	64
POOL-2	12×12×64	0	0
FLATTEN	9216×1	0	0
FC-3	3×1	9216×3	3

b) 请根据上表的网络结构用 Pytorch 实现(Activation function 用 ReLU)

对应源码在 jupyter nootbook 中，额外给出了一个随机化样本输入，验证了网络结构的正确性。

3，下面是一段摘自 Wikipedia 关于 Variational autoencoder 的描述：

From a formal perspective, given an input dataset X characterized by an unknown probability distribution $P(X)$, the objective is to model or approximate the data's true distribution $\boldsymbol{P}$ using a parametrized distribution P_θ having parameters θ. Let Z be a random vector jointly-distributed with X. Conceptually, Z will represent a latent encoding of X. Marginalizing over Z gives

$$P_\theta(X) = \int_Z P_\theta(X, Z) dZ$$

, where $P_\theta(X, Z)$ represents the joint distribution under θ of the observable data X and its latent representation or encoding Z. According to the chain rule, the equation can be rewritten as

$$P_\theta(X) = \int_Z P_\theta(X, Z) dZ = \int_Z P_\theta(X|Z) P_\theta(Z) dZ.$$

In the vanilla variational autoencoder, Z is usually taken to be a finite-dimensional vector of real numbers, and $P_\theta(X|Z)$ to be a Gaussian distribution. Then $P_\theta(X)$ is a mixture of Gaussian distributions.

It is now possible to define the set of the relationships between the input data and its latent representation as follows:

$$\text{Prior } P_\theta(Z)$$

$$\text{Likelihood } P_\theta(X|Z)$$

$$\text{Posterior } P_\theta(Z|X)$$

Unfortunately, the computation of $P_\theta(X)$ is expensive and in most cases intractable. To speed up the calculus to make it feasible, it is necessary to introduce a further function to approximate the posterior distribution as

$$Q_\Phi(Z|X) \approx P_\theta(Z|X)$$

with Φ defined as the set of real values that parametrize Q.

In this way, the overall problem can be easily translated into the autoencoder domain, in which the conditional likelihood distribution $P_\theta(X|Z)$ is carried by the probabilistic decoder, while the approximated posterior distribution $Q_\Phi(Z|X)$ is computed by the probabilistic encoder.

请根据以上描述求解 Variational autoencoder 的 ELBO loss function（请使用描述中的 notation 写出具体计算过程）？并解释为什么优化 ELBO loss function 能够“Maximize Likelihood”？

对应 markdown 和 latex 源码在 jupyter nootbook 中，此处就以图片的形式展示啦

1. **ELBO loss function**

解： ELBO 也即 evidence lower bound,也称 variational lower bound。

可以认为是在Q_Φ不断优化过程中（Q和P越来越接近），VAE导出的likelihood的下界

此时模型进行变分推断，使用高斯混合分布 和 EM算法

- 预估高斯混合分布各参数取值
- 在此高斯混合分布上进行采样，产生新的样本

step1：原样本x在通过encoder后，依据两输出$\mu'(x), \sigma'(x)$ sample出z

step2:此后需要把z过decoder，生成$\mu(z), \sigma(z)$来重新产生x，使得在高斯混合分布上sample出的x的似然最大，进而确定x所处的具体正态分布参数$\mu(z), \sigma(z)$。

也即

$$\max L = \max \sum_x \log P_\theta(x)$$

以下先给出ELBO loss function的代数推导

$$\begin{aligned}
\log P_\theta(x) &= \log P_\theta(x) \cdot \int_z P_\theta(z|x) dz \\
&= \log P_\theta(x) \cdot \int_z Q_\Phi(z|x) dz \\
&= \int_z Q_\Phi(z|x) \cdot \log P_\theta(x) dz \\
&= \int_z Q_\Phi(z|x) \cdot \log\left(\frac{P_\theta(z,x)}{P_\theta(z|x)}\right) dz \\
&= \int_z Q_\Phi(z|x) \cdot \log\left(\frac{P_\theta(z,x)}{Q_\Phi(z|x)} \cdot \frac{Q_\Phi(z|x)}{P_\theta(z|x)}\right) dz \\
&= \int_z Q_\Phi(z|x) \cdot \log\left(\frac{P_\theta(z,x)}{Q_\Phi(z|x)}\right) + \int_z Q_\Phi(z|x) \log\left(\frac{Q_\Phi(z|x)}{P_\theta(z|x)}\right) \\
&= \int_z Q_\Phi(z|x) \cdot \log\left(\frac{P_\theta(z,x)}{Q_\Phi(z|x)}\right) + KL\left(Q_\Phi(z|x) | P_\theta(z|x)\right) \\
&= \mathcal{L}(\theta, \Phi, x) + KL\left(Q_\Phi(z|x) | P_\theta(z|x)\right)
\end{aligned}$$

> 其中$\mathcal{L}(\theta, \Phi, x)$即为需要简化的ELBO loss,化简过程如下
>
> $$\begin{aligned}
> \mathcal{L}(\theta, \Phi, x) &= \int_z Q_\Phi(z|x) \cdot \log\left(\frac{P_\theta(z,x)}{Q_\Phi(z|x)}\right) \\
> &= \int_z Q_\Phi(z|x) \cdot \log\left(\frac{P_\theta(x|z) \cdot P_\theta(z)}{Q_\Phi(z|x)}\right) \\
> &= \int_z Q_\Phi(z|x) \cdot \log\left(\frac{P_\theta(z)}{Q_\Phi(z|x)}\right) + \int_z Q_\Phi(z|x) \cdot \log(P_\theta(x|z)) \\
> &= -KL(Q_\Phi(z|x) || P_\theta(z)) + \mathbb{E}_{z \sim Q_\Phi(z|x)}[\log P_\theta(x|z)]
> \end{aligned}$$
>
> 此式即为化简后的ELBO loss

2. **为什么优化ELBO loss function能够“Maximize Likelihood”?**

在推导过程中，可以发现引入的Q_Φ可以是任意函数

- Q_Φ的引入不会对$\log P_\theta(x)$的值造成影响
- Q_Φ的引入会同时影响$\mathcal{L}(\theta, \Phi, x)$和$KL\left(Q_\Phi(z|x) | P_\theta(z|x)\right)$

而由于

$$\log P_\theta(x) = \mathcal{L}(\theta, \Phi, x) + KL\left(Q_\Phi(z|x) | P_\theta(z|x)\right)$$

所以在只有Q_Φ变化的条件下，

$$\max_{Q_\Phi} \mathcal{L}(\theta, \Phi, x) \equiv \min_{Q_\Phi} KL\left(Q_\Phi(z|x) | P_\theta(z|x)\right)$$

所以由于KL散度的定义，当$Q_\Phi(z|x)$和$P_\theta(z|x)$尽可能接近相似时，$\mathcal{L}(\theta, \Phi, x)$和$\log P_\theta(x)$的函数间隙越来越小，也就可以认为

$$\max_{Q_\Phi} \mathcal{L}(\theta, \Phi, x) \equiv \max \log P_\theta(x)$$

也即优化ELBO loss function能够“Maximize Likelihood”