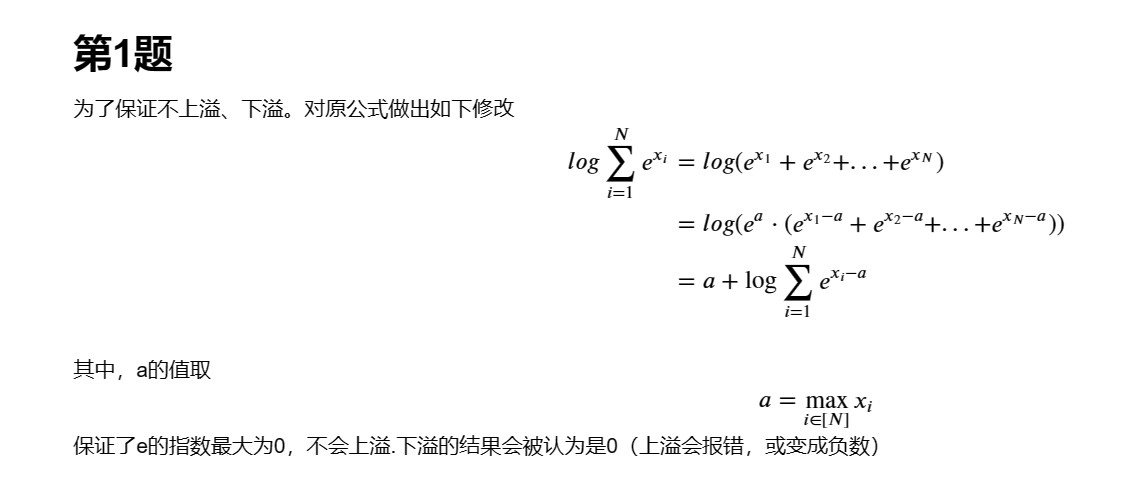
作业一

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**



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

1. 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 |

1. 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 ***P*** using a parametrized distribution having parameters . Let *Z* be a random vector jointly-distributed with *X*. Conceptually, *Z* will represent a latent encoding of *X*. Marginalizing over *Z* gives

, where represents the joint distribution under of the observable data and its latent representation or encoding . According to the chain rule, the equation can be rewritten as

In the vanilla variational autoencoder, is usually taken to be a finite-dimensional vector of real numbers, and to be a Gaussian distribution. Then 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:

Prior

Likelihood

Posterior

Unfortunately, the computation of 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

with defined as the set of real values that parametrize .

In this way, the overall problem can be easily translated into the autoencoder domain, in which the conditional likelihood distribution is carried by the probabilistic decoder, while the approximated posterior distribution 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**

