CS 450 Numerical Analysis (22Fa): Homework 1

Xu Ke (3190110360)

Oct. 10, 2022

1 Question 1: Briefly describe an area of interest where numerical computation is relevant. What kinds of mathematical problems are involved? What practical concerns, if any, are relevant (efficiency, accuracy and so on)?

Personally, my research interest lies in *Machine Learning* which becomes more and more popular nowadays to make the computers learn from the data and Numerical Analysis forms the foundation of many machine learning algorithms.

There are a certain number of theoretical ideas that have been developed with the objective of understanding modern deep learning methods, such as *Empirical risk minimization and empirical process theory, Implicit regularization, Linear regression in infinite dimension, Optimization and Generalization in the linear regime*.

Certain pairs of tradeoff are relevant, such as *Bias* (which indicates inaccuracy of the model prediction in comparison with the true value) and *Variance* (which indicates the change in target function if different training data is used), *Efficiency* (which can be defined as reducing the compute needed to train a specific capability) and *Accuracy* (which is an evaluation metric particularly used for classification tasks and represents the percentage of accurate predictions).

2 Question 2: Consider the infinite sum series $\sum_{n=1}^{\infty} \frac{1}{n}$ and answer the following questions:

2.1 Prove that the series is divergent.

I will use two intuitive idea to prove the divergence of harmonic series and credit given to This Lecture Notes.
First Proof: Suppose that the harmonic series converges with sum S. Then:

$$\begin{aligned} S &= 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + ... \\ &= (1 + \frac{1}{2}) + (\frac{1}{3} + \frac{1}{4}) + (\frac{1}{5} + \frac{1}{6}) + (\frac{1}{7} + \frac{1}{8}) + ... \\ &> (\frac{1}{2} + \frac{1}{2}) + (\frac{1}{4} + \frac{1}{4}) + (\frac{1}{6} + \frac{1}{6}) + (\frac{1}{8} + \frac{1}{8}) + ... \\ &= S \end{aligned} \tag{1}$$

The contradiction S > S concludes the proof.
Second Proof: There are 9 one-digit numbers, 1 to 9, whose reciprocals are greater than 1/10. Therefore:

$$H_9 > \frac{9}{10} \tag{2}$$

There are 90 two-digit numbers, 10 to 99, whose reciprocals are greater than 1/100. Therefore:

$$H_{99} > \frac{9}{10} + \frac{90}{100} = 2\frac{9}{10} \tag{3}$$

Continuing with this reasoning, it follows that:

$$H_{10^k-1} > k\frac{9}{10} \tag{4}$$

Since the subsequence $\{H_{10^k-1}\}$ is unbounded, the sequence $\{H_n\}$ diverges.

2.2 Explain why summing the series in floating point arithmetic yields a finite sum.

Actually this situation is caused by fixed point precision and floating point precision. There is a smallest number in the system, and once $\frac{1}{n}$ is less than that number, the "numerical harmonic series" no longer changes its value. In addition, there is a largest representable number ϵ_{mach} such that the numerical result of $\epsilon_{mach} + 1$ is still 1. Once $\frac{1}{n} < \epsilon_{mach}$, certainly the harmonic series up to $n-1$ will exceed 1 (since the first term is already 1), so again the series will no longer change. Thus, summing the series in floating point arithmetic yields a finite sum.

2.3 Would there exist a partial (but still infinite) sum of this series that converges? For instance, forms like $\sum_{nisaprimenumber} \frac{1}{n}$ or $\sum_{nisanoddnumber} \frac{1}{n}$. If yes, show an example; if no, give a proof.

Yes, an intuitive example would be $\sum_{nissquarenumber} \frac{1}{n}$. In this case, we can obviously re-write it as $\sum_{k\in N^+} \frac{1}{k^2}$. Then since

$$\sum_{n=1}^{N} \frac{1}{n^2} < 1 + \sum_{n=2}^{N} \frac{1}{n(n-1)} = 1 + \sum_{n=2}^{N}(\frac{1}{n-1} - \frac{1}{n}) = 1 + 1 - \frac{1}{N} \xrightarrow{N^{\infty}} 2 \tag{5}$$

2.4 Given any number a, if possible to construct a partial (but still infinite) sum of this series that converges to a? If yes, show how to do it.

Yes, adding the first n terms of the harmonic series produces a partial sum (harmonic number) and denote as $H_n = \sum_{k=1}^{n} \frac{1}{k}$. Obviously, these numbers grow very slowly, with logarithmic growth. More precisely, $H_n = lnn + \gamma + \frac{1}{2n} - \epsilon_n$ where $\gamma = 0.5772$ (Euler-Mascheroni constant) and $0 \leq \epsilon_n \leq \frac{1}{8n^2}$ which approaches 0 as n goes to infinity.

3 Question 3: Which of the following two mathematically equivalent expressions x^2-y^2 and $(x-y)(x+y)$ can be evaluated more accurately in floating-point arithmetic? Why? Moreover, for what values of x and y, relative to each other, is there a substantial difference in the accuracy of the two expressions?

It is more accurate to evaluate the expression as $\boxed{\text{(x-y)(x+y)}}$ in floating point system. intuitively, the expression $x^2 - y^2$ would exhibits more severe catastrophic cancellation if the absolute value of x and y are very close, for example $\boxed{\text{9000.2,y=9000.1}}$. Unlike the quadratic formula, $(x-y)(x+y)$ still has a subtraction, but it is a benign cancellation of quantities without rounding error, not a catastrophic. By theorem: *If x and y are floating-point numbers in a format with parameters β and p, and if subtraction is done with $p+1$ digits, then the relative rounding error in the result is less than* 2ϵ, the relative error in $x - y$ is at most 2ϵ and same is true of $x + y$. Multiplying two quantities with a small relative error results in a product with a small relative error.

More precise proof would be like this: Let's consider a double floating point system, each single elementary operation has a truncation error of at most half of unit in the last place, which is about a relative error of less than the fixed machine precision $\mu = 2^{-53}$. Then let's express the floating point realizations of the two expressions with relative errors $|\delta| \leq \mu$, we will get following results:

$$fl(fl(x^2) - fl(y^2)) = (x^2(1+\delta_1) - y^2(1+\delta_2)(1+\delta_3) = (x^2-y^2) + x^2\delta_1 - y^2\delta_2 + (x^2-y^2)\delta_3 + higher_order_terms \tag{6}$$

which has first order upper bound $[x^2 + y^2 + |x^2 - y^2|]\mu = 2max(x^2, y^2)\mu$

$$fl(fl(x+y)fl(x-y)) = ((x+y)(1+\delta_1)(x-y)(1+\delta_2))(1+\delta_3) \tag{7}$$

which has first order upper bound $3|x^2 - y^2|\mu$

4 Question 4: Read the paper "FP8 Quantization: The Power of the Exponent" and write a summary of the following:

4.1 What problem did this paper intend to solve, what are the critical challenges?

This paper in-depth investigates this benefit of the floating point format for neural network inference. From the academic point of view, many general models have very high indicators under the conventional 8bit quantization, so more and more papers have stepped into the 4bit stage, considering INT4 quantization on the general model. In the face of inevitable accuracy degradation, it is common to use weight or quantization parameters such as brecq, LSQ adjustment scheme. In industry, although most models can basically ensure accuracy under INT8 quantization, there will be many models with high accuracy requirements or special weight activation distribution, which will drop severely after INT8 quantization. The current state of the industry is that INT8 is barely usable, and in some cases requires higher bit configurations such as FP16, leaving INT4 out of the question. So is there a solution that can be in between int8 and FP16, and doesn't require a lot of tricks. Thus, it's easy to think of FP8.

4.2 How do the authors manage to tackle the challenge?

Firstly, the author introduces the representation mechanism of FP8 and the quantization effect of this number system on data. After introducing the advantages of FP8 to describe different distributions, the authors point out that the quantization parameter of FP8 should be obtained, which is similar to the quantization parameter scale of INT8, so that the precision of FP8 quantization can be simulated in software and the implementation of FP8 can be configured in hardware. In this paper, a simulation method of FP8 is proposed to find these parameters. Since FP8 quantization can be regarded as a uniform quantization grid of M-bits between successive integer powers, it means that we can simulate FP8 quantization of input vector x using the same method as simulated int8 uniform quantization. By referring to the solution of scale in int8, the optimal solution can also be divided into two ways: one is trained by QAT, and the other is PTQ. This paper also carries out rich experiments on different models using different 8bit quantization, showing the quantization results of several common models, and comparing the quantization results of QAT and PTQ.

4.3 What are the unique contribution of this paper?

This paper has shown that analytically the FP8 format can improve on the INT8 format for Gaussian distributions that are common in neural networks, and that higher exponent bits work well when outliers occur. This paper introduced a new way to simulate FP8 quantization in FP32 hardware that speeds up FP8 quantization simulation and makes it possible to learn the bias and mantissa-exponent bit-width trade-off. This paper validated the FP8 format for many networks in a post-training quantization setting, showing that generally for neural networks the 5M2E and 4M3E FP8 format works the best, and that for networks with more outliers like transformers increasing the number of exponent bits works best. This paper has also shown that when doing quantization-aware training, many of these benefits of the format disappear, as the network learns to perform well for the INT8 quantization grid as well.

4.4 What are the limitations? What's your idea to improve?

In this paper, they study the impact of the various FP8 formats on model accuracy, ignoring the specific impact of the hardware implementation on power consumption and latency. Evaluating the hardware impact difference between INT8 and FP8 is not easy for all networks. Both 8-bit formats incur similar overhead from a data transfer bandwidth perspective, and for computationally limited models where the relative overhead depends on the exact implementation, hardware design teams use accuracy analysis to make hardware trade-offs for their specific use cases and designs.

In addition, it is not clear whether the quantization of FP8 in the experiment will be per tensor or per channel, but the appendix states that if you apply per channel quantization, each channel will have its own maximum value of clip c, but the whole tensor will have a mantral number m.