CS 182: Project Initial Submission

Aviral Mishra, Nelson Lojo, Derrick Han Sun, Kesava Viswanadha

November 28 2023

1 Abstract

Improving inference time in transformers is an area of work that has seen great focus in their rise to fame. In particular, a number of methods have looked to approximating operations in the transformer architecture to reduce computation or improve inference time [3, 2, 5, 4]. The reduction of computation through approximation will be useful in cases where computational power is limited. We explore several possibilities for reducing computations, while attempting to minimize accuracy loss as a result of these shortcuts. We empirically show that our optimizations can reduce inference time while still approximating the attention mechanism and achieving comparable accuracy. Our code can be found at https://github.com/nelson-lojo/in-context-learning/tree/main.

2 Introduction

Transformers allow for the use of in-context learning, which requires the receiving of examples in the prompt. These examples are used to make a prediction for a new query input. This is one approach to allow models to adapt to a variety of tasks. However, not all systems are able to implement transformers due to computational costs or other conditional factors.

As a result of these limitations, we aim to test the feasibility of several optimizations:

1. *Combiners*[2] allow for full attention capability while maintaining subquadratic complexity via conditional expectation. Ren et al. have shown, within density estimation tasks, their capability to maintain accuracy while boosting speed.

2. Wortsman et al. have shown that for vision transformers, significant accuracy can be maintained when replacing the softmax activation function with ReLU after normalizing by the sequence length [3]. This results in a clear speed-up as ReLU does not require a gather step and is faster than computing an exponential (as in softplus, also explored in [3]).

3. Quantization-aware pretraining as in [5] has been shown to reduce both model size and speed up inference by replacing floating-point operations with Int8 operations.

4. Nyström approximations [4] seek to approximate attention by sampling rows and columns from full query and key matrices and performing smaller matrix multiplications to estimate the full attention matrix.

To test our optimizations, we will turn to the problem of learning a function class from in-context examples. The goal is to produce a model that not only can learn a function f from a function class F such that it can predict $f(x_{query})$ when given $x_1, f(x_1), x_2, f(x_2), ..., x_k, f(x_k), x_{query}$, but can also learn it quickly.

To test this, the functions f will be drawn from a distribution of functions D_F in F. Similarly, the x terms will be drawn i.i.d. from a distribution of inputs D_X. Models will be given the query $(x_1, f(x_1), x_2, f(x_2), ..., x_k, f(x_k), x_{query})$, and will seek to predict $f(x_{query})$. The error will be less than some amount, while the goal is to make predictions faster than the transformer design outlined in "What Can Transformers Learn In-Context? A Case Study of Simple Function Classes" [1]. Their transformer design will serve as our baseline. We will use several different function classes to test the effect of our optimizations on inference time and ability to make predictions with fewer in-context examples.

3 Training Methods for In-Context Learning

We will use the same decoder-only Transformer architecture as described in [1], but with our inference time optimizations implemented. To train our models, we will draw functions according to D_F as defined above, and inputs from D_X as defined above. Because we intend to use the the transformers from [1] as our baseline, we will use the same dimensions as them. This means that our inputs x will be of dimension $d = 20$. The inputs will be sampled from the isotropic Gaussian distribution $N(0, I_d)$. We will then compute each $y_i = f(x_i)$ and use them to produce the prompt $(x_1, y_1, ..., x_k, y_k, x_{query})$.

To evaluate the performance of our optimizations, we will analyze the time taken per prompt as a function of the number of in-context examples. This latency will be evaluated by comparing the optimized models against the model outlined in [1]. We will be doing this comparison using a variety of function classes: linear regression, linear classification, decision trees, and two-layer ReLU neural networks.

Although our primary goal is to improve the inference time, we will also be measuring accuracy. We will measure accuracy by the squared error as a function of amount of in-context examples, and we will compare this accuracy to the baseline model. Although we expect accuracy to decrease due to our optimizations, our secondary goal is to minimize this accuracy loss as much as possible.

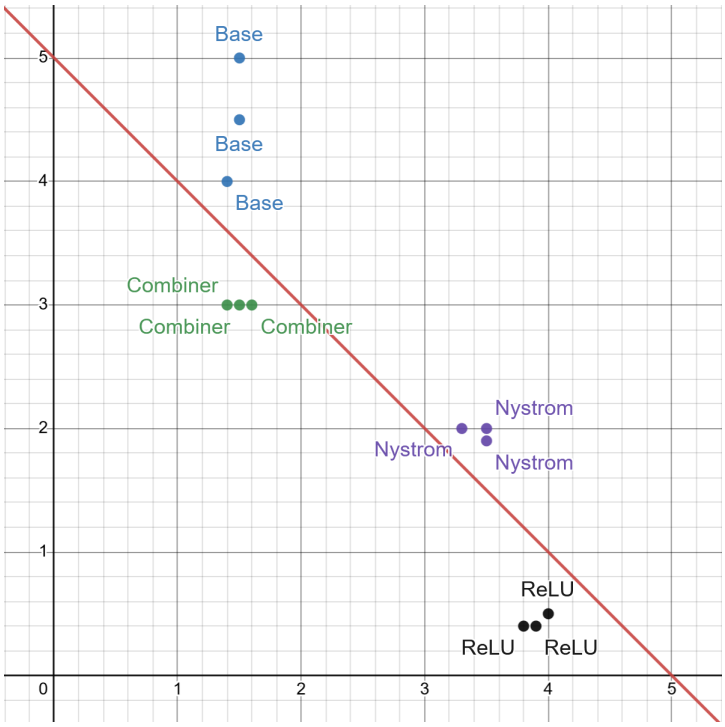

Figure 1: Mock-up plot of operation efficiency to accuracy

4 Results

We have implemented one out of our four proposed speedups and are executing training now. We ran into significant difficulties installing the packages used by [1]'s code in Google Colab and Kaggle Notebooks, and so are currently compute-limited to GitHub Codespaces. This initially led us to reimplement [1]'s code to not use the problem packages (`quinine` and `wandb`). However, we decided to remove as much as possible from their code when we realized we were unable to reach suitable progress. Due to the resulting compute constraint (training takes 100 hours in our GPU-less environment), we have not yet completed training to produce grounded plots. To illustrate our goal, we mimicked what we expect our plot of operation efficiency against accuracy to look like in figure 1 and how we expect our plot of in context examples to accuracy to look in figure 2.

5 Limitations and Future Exploration

This paper provided insights into the inference time speed-up of various optimizations for in-context learning of simple function classes. A notable limitation of this research is that it doesn't explore more complicated function classes.

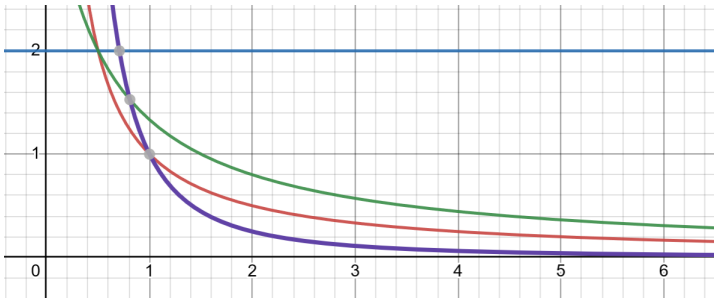

Figure 2: Mock-up plot of in-context examples to accuracy

Thus, further research could be done on the efficacy of the optimizations when in-context learning is performed for more complicated classes (e.g. Random Forests, 3+ layer Neural Networks).

Another limitation of this paper is the exploration of just four basic speed-up methods. Follow-up research could explore further optimizations that could surpass this paper's discussed methods in accuracy and/or speed-up.

The work done in this paper ties into the feasibility of transformers on less computationally capable hardware (e.g. the SCuM microchip), especially with the ReLU and quantization optimizations. Thus, additional research can be done with "small" transformers on less powerful chips.

6 Self Review

The main goal of the project is to speed up inference-time for in-context learning, and the main claims of this paper are that in-context learning speed-ups like quantization, Combiners, etc. may result in a significant improvement in inference time while retaining relatively good accuracy.

The experiments of this paper largely deal with training vanilla transformers to model a specific function class and benchmarking latency/accuracy, later applying one of their aforementioned inference speed-ups and re-benchmarking. This evaluation protocol particularly focuses on benchmarking, where accuracy is determined via squared loss and latency is determined via a simple time.clock() subtraction computation. Both are plotted against number of in-context examples provided, which allows for insight into how this variable intermingles with inference time speed.

The experiments would theoretically support the goal/claims of the paper if they produced graphs that indicated a clear drop in latency with the optimizations when compared to baseline, and if there wasn't a huge drop in accuracy to go along with it. However, because this paper currently doesn't have any real data yet, no statement can be made about the claims of this paper at the moment.

Limitations are discussed in the paper in the "Limitations and Future Ex-

ploration" section. These included the fact that the paper didn't explore more optimization methods and that more complicated function classes were not addressed.

The strengths of this paper is that it is very well organized, well-documented, and the training methods are explained very clearly which allows for reproducibility.

The major weakness of this paper are that there is no tangible data that is present yet, which clearly makes it very difficult to discuss results.

A suggestion for the improvement of this paper is to perhaps expand the scope of the topic to address some of the limitations that were mentioned before. For example, there could be an additional section of this paper dedicated to inferencing on complex function classes.

The relevant related works for this paper are as follows. "What Can Transformers Learn In-Context? A Case Study of Simple Function Classes" by Garg et. al forms the backbone of this research paper, as this paper reuses the code from the former heavily in order to establish a structure for training, function classes, and accuracy benchmarking. "Combiner: Full Attention Transformer with Sparse Computation Cost", "Replacing softmax with ReLU in Vision Transformers", "Nystromformer: A Nystrom-Based ALgorithm for Approximating Self-Attention", and "Q8BERT: Quantized 8Bit BERT" are the references for the optimizations that this paper ended up carrying out.

Because the processes of the experiments are very well-documented and descriptive, the experiments are able to be reran. However, no comments can be made about the reproducibility of the results themselves because no concrete results are present at the moment.

The plots in this paper (although they are just what we think we may get in temrs of data) are clearly interpretable because of the descriptive labelling of axes. The methodology behind the benchmarking is concisely and clearly explained in the "Training Methods for In-Context Learning".

The English in the paper is relatively correct and clear, other than slight grammatical errors and some excessive wordiness.

Much of the data this paper hopes to gather is still a TODO, which makes the critique of this paper very challenging. There is not much substance to this paper without lots of concrete data.

References

[1] Garg et al. "What Can Transformers Learn In-Context? A Case Study of Simple Function Classes" (https://arxiv.org/abs/2208.01066)

[2] Ren at al. "Combiner: Full Attention Transformer with Sparse Computation Cost" (https://arxiv.org/pdf/2107.05768.pdf)

[3] Wortsman et al. "Replacing softmax with ReLU in Vision Transformers" (https://arxiv.org/pdf/2309.08586.pdf)

[4] Xiong et al. "Nyströmformer: A Nyström-Based Algorithm for Approximating Self-Attention" (https://arxiv.org/abs/2102.03902)

[5] Zafrir et al. "Q8BERT: Quantized 8Bit BERT" (https://arxiv.org/pdf/1910.06188.pdf)