1 Bit Vit: Low-Bit Quantization Methods for Vision Transformers

Gautom Dos **Nicholas Henderson**

Sean Im

Vincent La **Ethan Lau**

1 Introduction and Motivation

The efficient deployment of neural networks has always been an important problem. With more companies developing large language models (LLMs) on huge, costly GPU clusters, research into how to adapt these models to run on consumer devices is a growing field. One new popular method to increase the efficiency of these LLMs is known as "quantization". This method has traditionally been applied to classic language-based LLMs. The goal of this project is to apply low-bit quantization to vision transformers.

Transformers, originally introduced by Vaswani et al. (2017), revolutionized natural language processing (NLP) by enabling machines to better understand and generate text through handling sequential data. This success in NLP led researchers to adapt transformers for computer vision, resulting in the development of vision transformers (ViTs). ViTs, as described by Dosovitskiy et al. (2021), apply the transformer architecture directly to images by treating them as sequences of patches. This approach demonstrated an alternative to conventional convolutional neural networks (CNNs) that also achieves great results and can sometimes be more efficient.

Quantization is a method that reduces the precision of a network's parameters to make the model smaller and faster. For example, if you had a model where all the weights were 16-bit floating numbers, quantizing it to be 8-bit integers would roughly half the size of the model and also allow for faster calculations. This is especially useful on devices with limited memory or compute power. The concept of quantization, particularly for efficient integer-arithmetic-only inference, was initially developed by Jacob et al. (2018) at Google. This study found it was possible to convert complex floating-point based models into simple integer-only ones that could run on mobile devices. Since then more recent developments such as GPTQ and BitNet have further researched the potential for significant model compression. GPTQ, for example, allows large GPT models to be compressed without a significant loss in performance, making them suitable for simpler hardware (Frantar et al., 2023).

The goal of this project is to apply low-bit quantization for language models to vision transformers. This could make vision transformers more energy-efficient and accessible for a wider range of devices, including those not traditionally equipped for advanced image recognition tasks. We plan on taking an existing vision transformer architecture, the ViT, and existing weights and attempt to quantize them using the methods prescribed in GPTQ and BitNet. Overall, this project can have applications in various sectors where efficient AI makes it more accessible to more users.

2 Relevant Papers and Codebases

2.1 4-bit Quantization

The foundational work on 4-bit quantization of language models provides valuable insights into maintaining accuracy under significant compression. This proposal envisions a similar leap for ViTs, utilizing the principles that have guided the successful quantization of language models. The goal is to achieve a balance where the efficiency gains from quantization do not come at the expense of the model's performance in image recognition tasks.

The "Towards Accurate Post-Training Quantization for Vision Transformer" study by Yifu Ding et al. is particularly instructive for our endeavor. It offers a novel approach to post-training quantization that significantly boosts ViTs' performance, especially in lower bit-width settings. This framework, with its innovative calibration scheme and attention to the Softmax function's integrity, serves as a blueprint for our project. It demonstrates the feasibility and benefits of applying low-bit quantization to ViTs, inspiring our approach to overcome the computational limitations we face.

The QLoRA codebase represents a significant advancement in model fine-tuning, enabling the training of large models on constrained resources. Its innovative techniques, such as the 4-bit NormalFloat (NF4) data type and Double Quantization, are particularly relevant to our project. Our realistic goal for this project is to utilize the advancements made by the QLoRA codebase and translate this to use for ViTs. The QLoRA codebase presents a way forward in making 4-bit quantization viable for transformers, and we hope to show its applicability in ViTs as well.

2.2 1-bit and 1.58-bit Quantization

On the more extreme side of quantization is binarization, or quantization down to a size of a single bit. Recently, binarization has been applied to large language models, specifically the transformer architecture. BitNet, introduced by Wang et al. (2023) has shown that binarization can be a fruitful approach for large language models. Experimental results from BitNet show that a 1-bit architecture can achieve competitive performance while reducing memory footprint and energy consumption even more than that of 8-bit or 4-bit quantization.

The authors utilize group quantization and normalization, which divides weights and activations into groups and then estimates estimate the parameters of each group independently. This allows their matrix multiplications to be partitioned across multiple devices since parameters can be computed locally, without further communication, which is important for scaling up large language models. Furthermore, the authors used mixed precision training, quanitizing weights and activations to a single bit but storing gradients and optimizer states in a high-precision format to ensure training stability and accuracy. For optimization, the authors suggest a large learning rate, since a small update often makes no difference for 1-bit weights. These findings will be important for us to consider, especially if we look to train a 1-bit ViT from scratch.

A PyTorch implementation of BitNet has also been published including a `BitLinear` module in their code repository, intended as a drop-in replacement for a `nn.Linear` layer. That being said, in order to use the `BitLinear` module we would have to train a ViT from scratch which we expect to be more difficult given our constraints on GPU resources.

Variants of this 1-bit architecture are being researched such as BitNet b1.58, introduced by Ma et al. (2024). The authors here allow weights to take on 3 possible values -1,0,1 as opposed to just 2 possible values in BitNet. The authors have also found that Bitnet b1.58 retains a lot of the benefits of BitNet in terms of memory and energy consumption. An additional advantage is stronger modeling capability with the inclusion of 0 in the model weights which supports explicit support for feature filtering. Their work demonstrates a new scaling law with respect to model performance and inference cost between their quantized model and full precision models.

3 Expected Plans and Evaluation Protocols

3.1 Realistic Plan

Our objective is to implement 4-bit quantization for Vision Transformers, drawing on the successes and strategies from quantization in both the APQ-ViT framework and the QLoRA codebase. This involves customizing these techniques to suit the unique architecture and requirements of ViTs, ensuring that we don't sacrifice the model's efficacy in image recognition, even as we push for greater efficiency and compactness.

By adopting 4-bit quantization for Vision Transformers, this project aims to not only enhance the computational efficiency of these powerful models but also expand their applicability. In doing so, we aspire to contribute to the broader adoption of efficient AI models across various industries, making high-performance image recognition more accessible and sustainable.

3.2 Moonshot Plan

Our moonshot objective would be implementing 1-bit or 1.58-bit quantization for vision transformers similar to BitNet and BitNet b1.58 for large language models. We anticipate these tasks to be more difficult since we would have to train a vision transformer from scratch for these quantization methods as opposed to a post-training quantization. A strong codebase for BitNet is available which looks promising but will likely still need to be adapted to fit the architecture of a vision transformer. Additionally, work on implementing BitNet b1.58 is still currently being done so wouldn't have as much to work with for that method. An application of our work that we would look to implement would be a simple browser and laptop demo with a quantized vision transformer, demonstrating an image classification task in real-time.

3.3 Evaluation Protocol

Because of the power of vision transformers to scale better than convolutional neural networks (CNNs) in vision tasks with more data, there has been a plethora of research and variations of the original architecture. For simplicity's sake, we chose to optimize the original vision transformer architecture that came out with the "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale" paper by Dosovitskiy et al. from Google (2021). The architecture is relatively simple. Input images are divided into patches and embedded with positional information. These embeddings are fed into a transformer encoder and subsequent MLP head, which predicts class probabilities.

Since we are optimizing an existing model architecture, evaluating any performance gains will be important. Quantization brings numerous benefits, but the most obvious and quantifiable are memory savings while using the model and lower computation time during inference. For our foundation, we will use a translation of the original ViT architecture in PyTorch, as the original model was implemented in JAX/Flax. Although we could have used the original codebase from Google, an implementation in PyTorch means that we can easily transfer established methods from QLoRA, GPTQ, and BitNet, all of which were written using PyTorch. There is also the benefit that the PyTorch implementation is compatible with ViT-Base and ViT-Large pre-trained weights from Google, meaning that we will not have to train from scratch, something that we do not have the resources to do.

Once we train a quantized ViT, we must evaluate the model. Before determining any memory or latency gains, we will need to evaluate the model's accuracy. To do so, we can look back to the original ViT paper. The paper gives a table containing baseline accuracies for the CIFAR-10 and CIFAR-100 datasets for each of their model types. To evaluate our model's accuracy, we will test it against the CIFAR-10 and CIFAR-100 datasets and see how it compares to the reported accuracies from the paper. In addition, we can use this test as the front for our memory and latency tests. By running the model as is with the pre-trained weights from Google, we can generate a baseline for memory usage and inference latency on the CIFAR-10 and CIFAR-100 datasets. We would effectively be killing two birds with one stone.

The performance measurement of a GPU program involves kernel execution time, computational efficiency, memory usage, and energy consumption. If we plan to use Nvidia's CUDA, we could use Nvidia's performance tools such as Nsight Systems and Nsight Compute which allows detailed measurement of usage of compute units, L1/Texture cache, L2 cache, and DRAM.

If we happen to have to handle bit level operations ourselves or integrate another codebase, it will be necessary to be mindful of memory alignment of < 1 byte data. Although the performance gain for memory usage is more apparent, implicit type conversion on a modern compiler and hardware is something to be careful of as "only quantizing the model weights without computing in lower-bit data types (i.e., keeping activation in FP16 or FP32) introduces no latency improvement (or even slower due to type conversion at runtime) but only memory saving" (Wu et al 2023). As CUDA provides no native support of a data type less than 8 bits, we need to search for a suitable module for our task.

As we only plan to change the nn.Linear() with a faster BitLinear() function in an existing ViT codebase, most of our efforts will be on comparing performance of these two functions. At the macro-level, after we are done with implementing BitLinear(), analysis of the new performance bottleneck in the ViT pipeline and scaling study will be necessary. The BitNet b1.58 paper also promises improvement in scalability of latency and memory usage as the model size increases. For the scaling study, we may have to measure performances at different model sizes to see if it scales better than a control model.

The performance measurement of a GPU program involves kernel execution time, computational efficiency, memory usage, and energy consumption. If we plan to use Nvidia's CUDA, we could use Nvidia's performance tools such as Nsight Systems and Nsight Compute which allows detailed measurement of usage of compute units, L1/Texture cache, L2 cache, and DRAM.

4 Conclusion

The quest for efficiency in artificial intelligence deployments, especially in neural network applications, is more pressing than ever. With the advent of ViTs as a formidable architecture for image recognition, leveraging their potential without the constraints of high computational demands becomes crucial. This proposal seeks to bridge this gap by adopting 4-bit quantization techniques, a strategy proven effective in compressing large language models without sacrificing significant performance. By extending these techniques to ViTs, we aim to unlock new possibilities for their application across a variety of systems, including those with limited computational resources.

Quantization's role in model compression and efficiency enhancement cannot be overstated. It offers a pathway to transform the traditionally resource-intensive ViTs into more accessible, faster, and compact models. The journey of quantization in the NLP domain, particularly the success stories of BitNet and GPTQ, highlights its potential to maintain model integrity while drastically reducing size. This project is anchored in the belief that such transformative quantization techniques can be adapted for ViTs, paving the way for their wider application in fields desiring efficient AI systems.

5 References

[1] Artidoro. (2023, July 19). Artidoro/Qlora: Qlora: Efficient finetuning of quantized llms. GitHub. https://github.com/artidoro/qlora

[2] Ding, Y., Qin, H., Yan, Q., Chai, Z., Liu, J., Wei, X., Liu, X. (2023, March 25). Towards accurate post-training quantization for vision transformer. arXiv.org. https://arxiv.org/abs/2303.14341

[3] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv preprint arXiv:2010.11929.

[4] Frantar, E., Ashkboos, S., Hoefler, T., Alistarh, D. (2023). GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. arXiv preprint arXiv:2301.12345.

[5] Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D. (2018). Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. arXiv preprint arXiv:1712.05877.

[6] Jeonsworld. (n.d.). Jeonsworld/VIT-pytorch: Pytorch reimplementation of the Vision Transformer (an image is worth 16x16 words: Transformers for image recognition at scale). GitHub. https://github.com/jeonsworld/ViT-pytorch

[7] Ma, S., Wang, H., Ma, L., Wang, L., Wang, W., Huang, S., … Wei, F. (2024). The Era of 1-bit LLMs: All Large Language Models are in 1.58 Bits. arXiv [Cs.CL]. Retrieved from http://arxiv.org/abs/2402.17764

[8] Wang, H., Ma, S., Dong, L., Huang, S., Wang, H., Ma, L., … Wei, F. (2023). BitNet: Scaling 1-bit Transformers for Large Language Models. arXiv [Cs.CL]. Retrieved from http://arxiv.org/abs/2310.11453

[9] Wang, H. (n.d.). BitNet. GitHub. https://github.com/kyegomez/BitNet

[10] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017).

[11] Wu, X., Li, C., Reza Yazdani Aminabadi, Yao, Z., He, Y. (2023). Understanding INT4 Quantization for Transformer Models: Latency Speedup, Composability, and Failure Cases. https://doi.org/10.48550/arxiv.2301.12017