Modeling and Scaling of Generative AI Systems

Project proposal

Aria David Darmanger[1], Owen Gombas[1], and Arthur Gygax[2]

[1] *University of Bern*
University of Neuchâtel
University of Fribourg
Swiss Joint Master of Science in Computer Science

28.10.2024

1. Project Proposal

1.1 Problem description

The aim of the project is to optimize the BERT model for text generation by applying pruning and quantization techniques.

Pruning is a technique used in neural networks to reduce model complexity by removing redundant or less critical parameters, such as certain neurons or connections, which do not significantly impact model performance. It can be applied during or after training and aims to minimize the model's size, reducing computation needs without major accuracy loss.

Quantization is a model compression technique that optimizes neural networks by reducing the precision of weights and activations from higher to lower bit representations, such as from 32-bit floating point to 8-bit integers. Often applied after training (post-training quantization), it reduces the memory footprint of models, making them more efficient and faster for inference.

We will focus on balancing the trade-off between performance in term of response time and quality of the generated text to evaluate the results before and after applying optimization techniques. The goal is to reduce model size and computational load while maintaining an acceptable level of accuracy and text quality.

1.2 Experimental Setup

The experimental setup consists of four main phases aimed at optimizing BERT for a text generation task in a chatbot environment with a focus on execution efficiency and response quality.

Phase 1: Baseline Model Setup
We will fine-tune a pre-trained BERT model for text generation using the Hugging Face Transformers library, specifically to adapt BERT for coherent and contextually accurate responses. Initial evaluations will capture baseline metrics, including execution time, response time, BLEU and ROUGE.

Phase 2: Model Pruning
To reduce model size and improve inference speed, we will apply magnitude-based pruning to the fine-tuned model using PyTorch's pruning. This will involve removing weights with low magnitudes or applying structured pruning to prune entire attention heads or layers. Key metrics will include model size reduction, execution time, response time, and text quality (BLEU/ROUGE/Perplexity), aiming for minimal quality degradation (<5%). Pruning levels (e.g., 30%, 50%) will be evaluated for their impact on these metrics.

Phase 3: Model Quantization
Next, we will apply post-training quantization and quantization-aware training to reduce the precision of the model's weights (e.g., from FP32 to FP16 or INT8) using PyTorch quantization utilities. Quantization will target size reduction and speed improvements, particularly beneficial for mobile or low-power deployment. We will measure metrics such as model size, execution time, energy consumption, and text quality to evaluate any trade-offs introduced by quantization.

Phase 4: Performance Evaluation and Optimization
In the final phase, we will iteratively optimize the model by combining pruning and quantization. We will experiment with different combinations (e.g., 30% pruning with INT8 quantization) to assess the best balance of model performance and quality. Statistical analysis, including Design

of Experiments (DoE) and ANOVA, will be used to evaluate how parameters like pruning ratios, quantization levels, and batch sizes affect performance metrics. Queueing theory simulations will model real-time response performance under concurrent user loads, with the goal of achieving an optimal response time.