Course: Independent study

MODEL COMPRESSION ON LARGE LANGUAGE MODELS

May 12, 2024

Name :- Abhinav Anand
Roll :- 2020101054

1 Introduction

Large Language Models (LLMs) have revolutionized natural language processing by achieving remarkable performance across a wide range of tasks, from language generation to understanding and translation. However, their success comes with a significant computational cost, both in terms of memory and processing power. As LLMs continue to grow in size and complexity, their deployment and practical use become increasingly challenging, particularly in resource-constrained environments.

To address these challenges, researchers and practitioners have developed a variety of model compression techniques tailored specifically for LLMs. These techniques aim to reduce the size and computational complexity of LLMs while preserving their performance, thus making them more accessible and efficient for deployment in real-world applications. Some of the key model compression techniques [3] include:

- **Pruning** Pruning involves removing redundant or less important parameters from the model, thereby reducing its size while minimizing the impact on performance.
- **Quantization** Quantization reduces the precision of the model's parameters, typically from floating-point numbers to lower-bit fixed-point numbers, leading to smaller memory footprint and faster inference.
- **Weight Sharing:** Weight sharing techniques aim to reduce the number of unique parameters in the model by sharing weights across different parts of the network, thereby reducing redundancy and saving memory.
- **Sparsity Regularization:** Sparsity regularization encourages certain parameters in the model to become zero or close to zero during training, resulting in a sparse model with fewer non-zero parameters.
- **Knowledge Distillation** Knowledge distillation involves training a smaller, more compact model to mimic the behaviour of a larger, pre-trained model, thereby transferring the knowledge learned by the larger model to the smaller one.
- **Low-Rank Factorization:** Low-rank factorization techniques decompose the weight matrices of the model into low-rank matrices, leading to a more compact representation of the model with fewer parameters.

1.1 Motivation

The motivation behind exploring model compression techniques in Large Language Models (LLMs) for summarization and translation tasks stems from several key factors:

- **Scalable Solutions:** The demand for efficient Large Language Models (LLMs) arises from the necessity for scalable natural language processing solutions capable of handling diverse platforms and environments.

- **Real-world Impact:** LLMs are integral to applications such as information retrieval, content generation, and cross-lingual communication, driving the need for efficient models in practical settings.
- **Environmental Concerns:** Given the substantial computational resources required by LLMs, there's a growing need to mitigate their environmental impact by reducing energy consumption and carbon emissions.

1.2 Challenges

Despite the compelling motivation to explore model compression techniques in LLMs for summarization and translation tasks, several challenges must be addressed:

- **Semantic Preservation:** Maintaining semantic richness while compressing LLMs is crucial for ensuring the accuracy and coherence of generated summaries and translations.
- **Trade-off Management:** Balancing the reduction in model size with the preservation of performance metrics poses a significant challenge, requiring careful optimization and evaluation.
- **Task-specific Adaptation:** Summarization and translation tasks have distinct requirements, necessitating tailored compression techniques to ensure optimal performance across different applications.
- **Computational Overheads:** While model compression aims to reduce computational costs, the compression process itself may introduce additional overheads, requiring effective management for practical deployment.

Throughout this report, we will delve into the principles underlying each compression technique, explore their implementation in the context of LLMs for summarization and translation tasks, and conduct a comprehensive comparative analysis. By evaluating the impact of each technique on model size, computational efficiency, and task performance, we aim to provide insights into their relative strengths and limitations, ultimately guiding the selection of the most suitable compression approach for practical deployment in real-world applications.

2 Problem Statement

This project sets out to comprehensively explore and assess various model compression methodologies within the framework of Large Language Models (LLMs), specifically focusing on their application in tasks such as summarization and translation. The primary objective is to delve into techniques like pruning, quantization, and knowledge distillation,

dissecting their efficacy and implications for optimizing LLM performance in these critical language processing endeavors.

3 Dataset

We have used CNN/Daily Mail for the summarization tasks and WMT 14 English-German Dataset for the translation tasks.

CNN/Daily Mail is a dataset for text summarization. Human-generated abstractive summary bullets were generated from news stories on CNN and Daily Mail websites as questions (with one of the entities hidden) and stories as the corresponding passages from which the system is expected to answer the fill-in-the-blank question. The corpus has 286,817 training pairs, 13,368 validation pairs and 11,487 test pairs, as defined by their scripts. The source documents in the training set have 766 words spanning 29.74 sentences on average, while the summaries consist of 53 words and 3.72 sentences. We have used only 10% of the original training set for training.

WMT 2014 English-German: Comprises high-quality news articles from Europarl, News Commentary, and TED Talks, offering realistic and diverse text domains. The corpus has 445,7749 training pairs, 2,994 validation pairs and 3,003 test pairs. We have used only 10% of the original training set for training.

4 Methodology and Evaluation:

4.1 Models:

We used 2 model, BART for Machine Translation Task and T5 for summarization task.

BART[1] is a denoising autoencoder for pretraining sequence-to-sequence models. It is trained by (1) corrupting text with an arbitrary noising function and (2) learning a model to reconstruct the original text. It uses a standard Transformer-based neural machine translation architecture. It uses a standard seq2seq/NMT architecture with a bidirectional encoder (like BERT) and a left-to-right decoder (like GPT).

T5[2] or Text-to-Text Transfer Transformer, is a Transformer based architecture that uses a text-to-text approach. Every task – including translation, question answering, and classification – is cast as feeding the model text as input and training it to generate some target text. This allows for the use of the same model, loss function, hyperparameters, etc. across our diverse set of tasks.

4.2 Model Compression Techniques:

Model compression techniques aim to reduce the size and computational complexity of Large Language Models (LLMs) while preserving their performance on natural language processing tasks such as summarization and translation. We explored three prominent

model compression techniques and their application in optimizing LLMs for efficient deployment:

Pruning: It involves the systematic removal of redundant connections or parameters from the model, resulting in a more compact architecture. In the context of LLMs, pruning targets weights or connections that contribute minimally to the model's overall performance. By selectively removing these parameters, pruning creates sparse networks that retain the essential information necessary for accurate predictions. Common pruning techniques include:

- **Magnitude-based Pruning:** Parameters with small magnitudes or gradients are pruned based on predefined thresholds.
- **Structured Pruning:** Entire neurons, layers, or sub-networks are pruned to achieve higher compression rates while maintaining structural integrity.
- **Iterative Pruning:** Pruning is performed iteratively, with retraining cycles to fine-tune the remaining parameters and mitigate performance degradation.

Quantization: It aims to reduce the precision of model parameters, such as weights and activations, by representing them with a lower number of bits. In LLMs, quantization leverages the observation that high precision may not be necessary for all parameters, allowing for significant reductions in memory footprint and computational requirements. Common quantization techniques include:

- **Fixed-point Quantization:** Parameters are quantized to a fixed number of bits, typically 8-bit or lower, reducing memory usage and enabling faster computation.
- **Dynamic Quantization:** Quantization thresholds are dynamically adjusted during inference based on the distribution of parameter values, optimizing precision without sacrificing accuracy.
- **Vector Quantization:** Clustering techniques are applied to group similar parameter values, reducing the number of unique representations required.

Knowledge Distillation: Knowledge distillation involves transferring knowledge from a large, cumbersome model (the teacher) to a smaller, more lightweight model (the student). In the context of LLMs, knowledge distillation enables the creation of efficient student models that can match or even surpass the performance of their larger counterparts. Common knowledge distillation techniques include:

- **Soft Target Training:** The teacher model's soft predictions, i.e., probability distributions over classes, are used as training targets for the student model, enabling finer-grained information transfer.

- **Feature Mimicking:** The student model is trained to mimic the intermediate representations or features learned by the teacher model, capturing higher-level abstractions without directly replicating the teacher's predictions.

- **Attention Distillation:** Attention maps generated by the teacher model are used to guide the training of the student model, improving its ability to focus on relevant input tokens and contexts.

4.3 Evaluation:

We will automatically evaluate our methods on ROUGE metrics for summarization task and .

- **ROUGE-N:** measures the number of matching 'n-grams' between our model-generated text and a 'reference'.

- **ROUGE-L:** measures the longest common subsequence (LCS) between our model output and reference. All this means is that we count the longest sequence of tokens that is shared between both.

- **BLEU:** measures the overlap in n-grams (sequences of n words) between the candidate translation and the reference translations. The BLEU score ranges from 0 to 1, with 1 indicating a perfect match between the candidate and reference translations.

5 Experiment

5.1 Summarization Task:

For summarization, we used the T5 model and evaluated the compression techniques on the Rouge score. Below are the implementations for each compression technique.

- In pruning, we applied unstructured L1 pruning with a pruning amount of 20% to all linear layers of the model.

- In quantization, we used fixed-point quantization; it does this by adding two special tools to the model, one for simplifying before the calculation(QuantStub) and another for turning the simplified answers back into accurate ones.

- In Knowledge distillation, we used T5-base as our teacher model, and T5-small as the student model, and the student model learns from the teacher model by mimicking its output.

5.2 Translation Task:

For Translation task, we used the BART model and evaluated the compression techniques on the BLEU score. Below are the implementations for each compression technique.

- In pruning, we applied unstructured L1 pruning with a pruning amount of 20% to all linear layers of the model.
- In quantization, we used 8-bit static quantization, which converts the model parameters and operations to quantized equivalent.
- In Knowledge distillation, we used BART-large as our teacher model, and BART-base as the student model, and the student model learns from the teacher model by mimicking its output.

6 Results and Analysis:

Table1 shows the performance of the T5 model and different versions of compression techniques on the summarization task. Compressing T5 models through pruning, quantization, or distillation leads to decreased text summarisation task performance. While all compressed models performed worse than the original T5 model, the distilled model suffered the most significant drop in quality as measured by ROUGE scores. The pruned model performs best in all compression techniques.

Methods	ROUGE-1	ROUGE-2	ROUGE-L
T5 model	45.02	23.6	37.03
Pruned T5 model	39.3	18.1	27.9
Quantized T5 model	39.2	17.8	27.8
Distilation T5 model	36.7	16.3	26.04

Table 1: Performance of T5 models on all compression techniques.

Table 2 shows the performance of the BART model and different versions of compression techniques on the machine Translation task. The table shows that the original BART model achieves the best BLEU score of 37.5. Among the compressed models, the Distilled BART model performs the closest to the original model, with a BLEU score of 36.3. The Pruned BART model and Quantized BART model achieve lower BLEU scores of 26.3 and 28.4, respectively. Overall, the results in the table suggest that compressing BART models using pruning or quantization techniques leads to a significant decrease in their performance on the BLEU metric, while distillation compression offers a better trade-off between model size and performance.

Methods	BLEU-Score
BART model	37.5
Pruned BART model	26.3
Quantized BART model	28.4
Distilation BART model	36.3

Table 2: Performance of BART models on all compression techniques.

7 Conclusion:

In this work, we have tried to address the problem of model compression techniques on different nlp tasks.Compressing large language models like T5 and BART using techniques like pruning, quantization, or distillation generally leads to a decrease in their performance on tasks like text summarization and machine translation. While compressed models offer a smaller size advantage, this comes at the cost of reduced accuracy as measured by metrics like ROUGE and BLEU scores. Among the compression methods, distillation appears to be the best compromise, offering a closer performance to the original model compared to pruning or quantization.

References

[1] Hugh A Chipman, Edward I George, and Robert E McCulloch. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.

[2] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[3] Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A survey on model compression for large language models. *arXiv preprint arXiv:2308.07633*, 2023.