Pruning OPT For Summarization

Andrew Zolensky, Chris Lee, Vinay Padegal, Sarang Sridhar

December 19, 2024

Abstract

Transformer-based Large Language Models (LLMs), such as OPT, LLaMA, and GPT-3, achieve state-of-the-art performance on natural language processing tasks but require significant computational and memory resources. In this study, we compress OPT models using pruning and task-specific fine-tuning, targeting summarization tasks. We evaluate the compressed models using both task-specific and resource-efficiency metrics. Our findings highlight the trade-offs between model accuracy and resource efficiency. This work provides insights into optimizing LLMs for deployment in constrained environments and establishes a framework for evaluating accuracy-resource trade-offs in model compression.

1 Introduction

Large language models (LLMs) have been shown to be effective in many natural language processing tasks. However, transformer-based LLMs such as OPT, Llama, and GPT3 require large amounts of computational and memory resources to train and perform inference tasks. Their large number of parameters (often to the tune of billions of parameters) preclude their effective deployment on mobile and other resource-constrained devices. Model compression and acceleration techniques such as pruning with fine-tuning and low rank factorization hold promise for making LLMs more resource-efficient. However, it is unclear to what extent successful compression and acceleration techniques can be composed.

We compose existing model compression and acceleration techniques in order to make OPT as small as possible, while retaining as much accuracy as possible. In order to understand the effects of doing this, we evaluate performance after various stages of compression, with a focus on both task-focused metrics and resource-focused metrics. In particular we evaluate several methods of pruning followed by fine-tuning to the particular task. We evaluate our model on a summarization task on the BILLSUM [Eidelman(2019)] dataset.

The input to our process is a Large Language (Embedding) Model, OPT. The output is compressed language model fine-tuned for a summarization task, where the weights have been pruned and fine-tuned. We do performance comparisons between models before and after key stages of the compression and acceleration process. These metrics include accuracy as well as resource-based metrics such as time spent training, memory consumed during training, and final model size and inference speed.

Our extensions are:

- Model contribution - We prune and fine-tune OPT 125M & 350M using various methods for summarization on the BILLSUM dataset
- Analysis contribution - We compare the performance of the fine-tuned models using both accuracy and resource-based metrics. We also do a qualitative analysis of the pruned models.

2 Literature Review

2.1 Wanda

[Sun et al.(2024)Sun, Liu, Bair, and Kolter] presents a novel pruning approach, called Wanda (Pruning by Weights and Activations), designed to induce sparsity in pre-trained large language models (LLMs). Wanda introduces two key innovations. The first is the Activation-Aware Scoring Metric: Wanda prunes weights by considering the smallest values obtained from the product of weights and input activations. This contrasts with methods like Magnitude Pruning, which rely solely on weight magnitudes and a set threshold. The researchers observed that in large LLMs (around 6 billion parameters or more), a small subset of hidden state features tends to have much larger magnitudes than others. This disparity, significant in larger LLMs, influences pruning effectiveness. Wanda leverages these prominent feature values alongside weight magnitudes to inform pruning. The second innovation is the Localized Comparison Group: Wanda calculates weight importance scores on a per-output basis, rather than on a layer-wide or network-wide basis. The team found that this localized comparison consistently yields better results than traditional layer-wise pruning methods. Wanda's performance is compared against Magnitude Pruning [Han et al.(2015)Han, Pool, Tran, and Dally] and SparseGPT [Frantar and Alistarh(2023)], where SparseGPT is a more complex approach involving weight updates and computationally demanding inverse Hessian calculations. Wanda, however, achieves similar efficiency without requiring gradient computations or weight updates, making it significantly faster than SparseGPT. Evaluated on

LLaMA and LLaMA-2 models, Wanda's performance is assessed using zero-shot tasks and language modeling. The results show Wanda substantially outperforms Magnitude Pruning and competes closely with SparseGPT. While pruned models show a gap in accuracy compared to dense models without fine-tuning, this gap reduces as model size increases. Notably, Wanda achieves unstructured 50% sparsity in LLaMA-65B and LLaMA-2-70B while maintaining zero-shot accuracy close to that of dense models.

2.2 SparseGPT

[Frantar and Alistarh(2023)] presents **SparseGPT**, an accurate one-shot pruning technique which works efficiently at the scale of models with 10-100+ billion parameters. The authors observe that the layer-wise pruning problem can be posed as the problem of finding, for each layer l, a sparsity mask M_l with a certain target density and possibly updated weights $\hat{W}_l$ such that

$$\arg \min_{\text{mask } \mathbf{M}_\ell, \widehat{\mathbf{W}}_\ell} \|\mathbf{W}_\ell \mathbf{X}_\ell - (\mathbf{M}_\ell \odot \widehat{\mathbf{W}}_\ell)\mathbf{X}_\ell\|_2^2. \tag{1}$$

They observe that finding the optimal solution to the above equation is NP-Hard and investigate a popular approach is to separate the problem into *mask selection* and *weight reconstruction*. In other words, if we first fix a pruning mask M, then the above optimization problem turns into a *linear squared error problem* that is easily optimized using the linear regression closed form solution at each layer i:

$$\mathbf{w}^i_{\mathbf{M}_i} = (\mathbf{X}_{\mathbf{M}_i}\mathbf{X}^\top_{\mathbf{M}_i})^{-1}\mathbf{X}_{\mathbf{M}_i}(\mathbf{w}_{\mathbf{M}_i}\mathbf{X}_{\mathbf{M}_i})^\top \tag{2}$$

However, this involves costly Hessian inverse calculations which make it impractical on models with the size of 10-100 billion parameters. The authors introduce a novel approximate sparse regression solver that performs these costly inverse Hessian calculations efficiently while retaining accuracy.

2.3 Finetuning

[Hu et al.(2021)Hu, Shen, Wallis, Allen-Zhu, Li, Wang, Wang, and Chen] presents a method which allows a user to indirectly finetune a Large Language Model by fine-tuning layer-specific rank-decomposition matrices instead, which are summed with the pre-trained weights of the LLM during inference. Rank-decomposition is a technique for factoring a matrix M $\in F^{n\times m}$ as $M = AB$ for $A \in F^{n\times r}$, $B \in F^{r\times m}$, where $r << n, m$, and F is an arbitrary field of dimension $n \times m$, such as $R^{n\times m}$. This equality can also be relaxed to an approximation. In LoRA, the practitioner fine-tunes an LLM by rewriting each layer, L, using $W^L_F = W^L_O +$

$A_L B_L$, where W_O^L is the original (frozen) weights matrix for layer L. During fine-tuning, only gradients for A_L and B_L are computed and applied. Inference uses the same formula. This makes training faster, by reducing the number of parameters to update. It also reduces model-adaptation to summation with low-rank matrices, which can be computed prior to deployment, leading to no additional memory footprint or inference latency. The approach is grounded in previous research which demonstrated that, while the gradient updates made to LLM weights during fine-tuning tend to have full rank, they also tend to have low "effective" rank, meaning that they are easily approximated via low-rank matrices. This makes some amount of mathematical sense from the perspective of the theory of random matrices. Random square matrices tend to have full rank, but also tend to be poorly conditioned. Since weights matrices are updated via summation with stochastic gradients ("random matrices"), this implies that we will often be adding poorly conditioned matrices to our weights matrices during training. Since poorly conditioned matrices are highly sensitive to slight changes in input, this introduces a high degree of "variance" into our models, which counteracts the desired "bias" we hope to be learning. As a result, we may be just as well, if not better-off training "low rank approximations" to the total weight updates, then adding these to the weights matrices of the pre-trained model. This strategy is "LoRA."

3 Experimental Design

3.1 Data

For our analysis, we utilize the **Billsum dataset** [Eidelman(2019)] from Hugging Face. The data consist of three parts: US bills (train set), US bills (validation set) and California bills (test set). The US bills were collected from the Govinfo service provided by the United States Government Publishing Office (GPO) under CC0-1.0 license. The California, bills from the 2015-2016 session are available from the legislature's website. The dataset is divided into three distinct splits: training, validation, and test. Below, we provide a summary of the number of examples in each split.

Table 1: Number of Examples in Each Data Split

Split	**Number of Examples**
Training	5,000
Validation	100
Test	100

We use the train split during the fine-tuning process and reserve the validation and test splits for benchmarking purposes. Note that the validation set could be considered out-of-distribution since it draws from a different source than the training set.

An example from the dataset is shown in Table 2.

Table 2: Example of a Bill and its Summary

Title: LIABILITY OF BUSINESS ENTITIES PROVIDING USE OF FACILITIES TO NONPROFIT ORGANIZATIONS.
Text: SECTION 1. LIABILITY OF BUSINESS ENTITIES PROVIDING USE OF FACILITIES TO NONPROFIT ORGANIZATIONS. (a) Definitions.–In this section: (1) Business entity.–The term "business entity" means a firm, corporation, association, partnership, consortium, joint venture, or other form of enterprise. (2) Facility.–The term "facility" means any real property, including any building, improvement, or appurtenance. (3) Gross negligence.–The term "gross negligence" means voluntary and conscious conduct by a person with knowledge (at the time of the conduct) that the conduct is likely to be harmful to the health or well-being of another person. (4) Intentional misconduct.–The term "intentional misconduct" means conduct .....
Summary: Shields a business entity from civil liability relating to any injury or death occurring at a facility of that entity in connection with a use of such facility by a nonprofit organization if: (1) the use occurs outside the scope of business of the business entity; (2) such injury or death occurs during a period that such facility is used by such organization; and (3) the business entity authorized the use of such facility by the organization. Makes this Act inapplicable to an injury or death that results from an act or omission of a business entity that constitutes gross negligence or intentional misconduct, including misconduct that.....

Below are example label summaries from the validation and test set that we will use to analyze the quality of the responses from the various trained models.

Example 1 (Validation set): "Existing law requires all moneys, except for fines and penalties, collected by the State Air Resources Board from a market-based compliance mechanism relative to reduction of greenhouse gas emissions to be deposited in the Greenhouse Gas Reduction Fund. Existing law establishes the Transit and Intercity Rail Capital Program, which receives 10% of the annual proceeds of the Greenhouse Gas Reduction Fund as a continuous appropriation, to

fund capital improvements and operational investments to modernize California's rail systems to achieve certain policy objectives, including reducing greenhouse gas emissions, expanding and improving rail services to increase ridership, and improving rail safety. Existing law",

Example 2 (Validation set): "Existing law provides that the Board of Parole Hearings or its successor in interest shall be the state's parole authority. Existing law requires that a prisoner who is found to be permanently medically incapacitated, as specified, be granted medical parole, if the Board of Parole Hearings determines that the conditions under which the prisoner would be released would not reasonably pose a threat to public safety. Existing law exempts a prisoner sentenced to death, a prisoner sentenced to life without the possibility of parole, and a prisoner who is serving a sentence for which parole is prohibited by initiative statute, from medical parole eligibility. Ex"

Example 3 (Test set):

To perform training on a decoder-only model, we structured the input training data in the following manner based on [Dassum(2024)]:

"Below is an instruction that describes a task. Write a response that appropriately completes the request.

Instruct: Summarize the text below.

[Insert text]

Output:

[Insert label summary]

End"

The QLoRA fine-tuning script was based on guides from [Dassum(2024)] and [HuggingFace(2024)]. Due to the input token limit of OPT models, we truncate the input text to 1024 tokens and the label summary to 128 tokens. When performing inference, we specify that the model should generate a maximum of 128 new tokens apart from the input prompt. We extract the text after the "### Output:" key and before the "### End" key to get the predicted summary.

For QLoRA fine-tuning, we specify a number of key parameters. We quantize the model to 8 bit. The QLoRA rank is set to 16, the scaling factor to 32, and the drop-out rate to 0.05. The target modules are the query, key, value projections and fully connected layers. There are no bias terms. We train for 3 epochs with a learning rate of 2e-4 using the Adam optimizer.

3.2 Evaluation Metric

- **ROUGE-L**: Rouge-L[Lin(2004)] is a recall-oriented metric that looks for the longest common subsequence between the reference and the candidate.

- **BERTScore**: BERTScore[Zhang et al.(2020)Zhang, Kishore, Wu, Weinberger, and Artzi], leverages contextual BERT embeddings to evaluate the semantic similarity of the generated and reference texts.
- **Resource-oriented Metrics:** We also calculate the number of parameters, time for inference on the test set and the runtime memory of the models.

3.3 Simple Baseline

For our simple baseline, we chose to use the base OPT-125m and 350m models (not fined-tuned). When performing inference on the validation and test datasets, we found that the OPT-125m and 350m models either generated blank outputs or repeated the input text up to the max token limit. In a few cases, the generated output differed from the input, but was only loosely related to the input text. For instance, in example 1, the generated output was "The State of California is committed to providing the best possible transportation services to the people of California", which was not the intent of the bill.

4 Experimental Results

4.1 Published Baseline

Our strong baseline consists of fully fine-tuned OPT-125m and 350m and QLoRA fine-tuned OPT-125m and 350m. As expected, the fully fine-tuned OPT models generated the best results. The generated outputs are all novel sentences that attempt to summarize. An example of a good generated summary for example 1 are as follows (see Data section for label summaries):

OPT-125m: "Amends the Public Resources Code to establish the Transit and Intercity Rail Capital Program to fund transformative capital improvements that will modernize California's intercity, commuter, and urban rail systems and ferry transit systems to achieve all of the following policy objectives: (1) reduce emissions of greenhouse gases; (2) expand and improve transit service to increase ridership; (3) improve transit safety; (4) improve transit service to increase ridership; (5) improve transit service to increase ridership; (6) improve transit safety; (7) enhance the connectivity, integration, and coordination of the state's various transit systems;"

OPT-350m: "Amends the Public Resources Code to create the Transit and Intercity Rail Capital Program to fund transformative capital improvements that will modernize California's intercity, commuter, and urban rail systems to achieve all of the following policy objectives: (1) reduce emissions of greenhouse gases; (2) increase ridership; (3) improve transit safety; (4) integrate the rail service of the

state's various rail operators; (5) improve transit safety; (6) enhance transit service; (7) improve transit safety; (8) increase ridership; (9) increase the capacity of an existing transit system; (10)"

Both models correctly summarized the bill's intent of funding rail improvements. However, note that both models repeat themselves after point (4).

The fully fine-tuned models can still fail to capture the essence of the text, as evidenced in example 2:

OPT-125m: "Amends the Penal Code to: (1) repeal the prohibition against the use of a sentence for which a prisoner is eligible for resentence or recall pursuant to subdivision (e); (2) provide that a prisoner is not eligible for resentence or recall pursuant to subdivision (a) if he or she was convicted of first-degree murder; (3) provide that a prisoner is not eligible for resentence or recall pursuant to subdivision (a) if he or she was convicted of first-degree murder or of first-degree murder; (4) provide that a prisoner who was convicted of first-degree murder or of first-"

OPT-350m: "Amends the Federal criminal code to revise the definition of "prisoner."

The output from OPT-125m is gibberish even by lawyer standards. It mentions resentencing and first-degree murder, which the bill does talk about, but in an ungrammatical manner and does not mention eligibility for medical parole, which was what the bill was mainly about. The output from OPT-350m is succinct and does allude to some kind of revision of imprisonment status but the bill was not trying to define what a prisoner is.

The fully fine-tuned models generated summaries more closely align with the label summaries for the test set. This is expected since the test set data draw from the same source as the training set. Consider example 3:

OPT-125m: "Special Agent Scott K. Carey Public Safety Officer Benefits Enhancement Act - Amends the Omnibus Crime Control and Safe Streets Act of 1968 to require an eligible dependent of a public safety officer to be eligible for assistance if such officer: (1) dies in the line of duty on or after January 1, 1978; or (2) becomes permanently and totally disabled as the direct result of a catastrophic injury sustained in the line of duty on or after January 1, 1978. Requires the applicant's spouse to be eligible for such assistance if the applicant's spouse dies or becomes permanently and totally disabled as the result of a catastrophic injury sustained in"

OPT-350m: "Special Agent Scott K. Carey Public Safety Officer Benefits Enhancement Act - Amends the Omnibus Crime Control and Safe Streets Act of 1968 to revise the eligibility requirements for educational assistance to include dependent children of a law enforcement officer who dies or becomes permanently disabled as the direct result of a catastrophic injury sustained in the line of duty on or after January 1, 1978. Allows eligible children of law enforcement officers

who die or become permanently disabled as the direct result of a catastrophic injury sustained in the line of duty on or after January 1, 1978, to: (1) receive assistance under the Omnibus Crime Control and Safe Streets Act of 1968"

Both models mentioned the law granting assistance to dependents of public safety officers who die or become disabled in line of duty.

The QLoRA fine-tuned OPT models are able to generate coherent summaries, but fall short of the accuracy achieved by the fully fine-tuned models. In addition, the 350m QLoRA model produces significantly better summaries than the 125m model. Consider example 1:

OPT-125m: "Amends the Penal Code to: (1) require a prisoner to be eligible for resentence or recall pursuant to subdivision (e) if he or she was convicted of first-degree murder if the victim was a peace officer, as defined in Section 830.1, 830.2, 830.3, 830.31, 830.32, 830.33, 830.34, 830.37, 830.38, 830.36, 830.37, 830.38, 830.39, 830.39, 830.40, 830."

OPT-350m: "Amends the Penal Code to require the head physician of an institution in which a prisoner is incarcerated to determine, as provided in this section, that the prisoner is permanently medically incapacitated with a medical condition that renders him or her permanently unable to perform activities of basic daily living."

The 125m model paraphrases the first part of the input text almost word-by-word and fails to mention anything medical parole related. The 350m model mentions medical incapacitation in the prison context as detailed in the bill but gets cut off by the token limit before mentioning medical parole; in other words, it wasn't succinct enough.

4.2 Extensions

As part of our extension, we explore three distinct pruning methods—*Wanda*, *Magnitude Pruning*, and *SparseGPT*—applied across three sparsity paradigms: *Unstructured Sparsity*, *2:4 Semi-structured Sparsity*, and *4:8 Semi-structured Sparsity*. Following the pruning process, we employ *QLoRa* fine-tuning to adapt each pruned model to the target dataset, ensuring performance alignment with downstream tasks.

Sparsity Techniques:

- **Unstructured Sparsity:** This technique prunes 50% of the weights on a per-output basis, without additional constraints. It provides flexibility in weight selection and generally results in better model performance compared to structured methods.
- **N:M Structured Sparsity:** This method also prunes 50% of the weights but

imposes the constraint that at most N out of every M contiguous weights remain non-zero.

  - **Benefits:** Models with structured sparsity can utilize sparse tensor cores for faster matrix multiplication.
  - **Trade-offs:** The constraints lead to a slight performance decline compared to unstructured sparsity, due to the restricted pruning process.

The following key observations can be drawn from the table:

- Performance Comparison: Models with unstructured sparsity generally outperform those with structured sparsity. For instance, Wanda with 50% unstructured sparsity achieves a Bertscore of 0.8616, which decreases to 0.8449 and 0.6088 for 4:8 and 2:4 structured sparsity, respectively. This aligns with our hypothesis that the fewer constraints in unstructured sparsity allow for better performance.
- Inference Speed: Structured sparsity offers faster inference times compared to unstructured sparsity. For example, the total inference time for unstructured sparsity is 1064.84 seconds, while it decreases to 1045.93 and 1038.63 seconds for 4:8 and 2:4 structured sparsities, respectively.

Table 3: Performance of Sparse Models Across Different Architectures

Model	Technique	Sparsity	Validation Data		Test Data		Time	Nonzero Parameters	Size (MB)
			Rouge	Bertscore	Rouge	Bertscore			
OPT 125m (Base)	-	-	0.1652	0.7243	0.0782	0.3627	278.42	125,239,296	955.66
OPT 125m (Full)	-	-	0.2338	0.8391	0.4239	0.8863	262.3	125,239,296	955.54
OPT 125m (QLoRa)	-	-	0.2274	0.8079	0.3729	0.8593	1073.65	126,124,032	484.64
OPT 125m (QLoRa)	Wanda	50%	0.1853	0.8232	0.3263	0.8616	1064.84	63,504,384	484.64
OPT 125m (QLoRa)	Wanda	4:8	0.2153	0.8263	0.2868	0.8449	1045.93	63,504,384	484.64
OPT 125m (QLoRa)	Wanda	2:4	0.2096	0.8107	0.1384	0.6088	1038.63	63,504,384	484.64
OPT 125m (QLoRa)	Magnitude	2:4	0.109	0.7771	0.2426	0.8203	1055.93	63,504,384	484.64
OPT 125m (QLoRa)	SparseGPT	2:4	0.1446	0.6918	0.1933	0.6528	1050.35	63,504,384	484.64
OPT 350m (Base)	-	-	0.0161	0.09	0.0127	0.0717	477.11	331,196,416	2526.92
OPT 350m (Full)	-	-	0.2602	0.8391	0.4158	0.8841	473.08	331,196,416	2526.92
OPT 350m (QLoRa)	-	-	0.2298	0.8413	0.397	0.879	2273.01	333,555,712	1281.69
OPT 350m (QLoRa)	Wanda	2:4	0.1415	0.791	0.2129	0.8141	2353.38	167,957,504	1281.69
OPT 350m (QLoRa)	Magnitude	2:4	0.0391	0.7814	0.0551	0.6537	2189.13	167,957,504	1281.69
OPT 350m (QLoRa)	SparseGPT	2:4	0.2395	0.8314	0.2994	0.8467	2271.95	167,957,504	1281.69

Importantly, the table above highlights the count of **non-zero** parameters. It is important to note that there is no *actual* reduction in the total number of parameters between the full and pruned models, as no entire channels of weights are removed. Instead, pruning methods simply set specific weights to zero, making the non-zero parameter count the key metric of interest.

Summarizing the results, OPT125m demonstrates strong performance in the summarization task based on BertScore, except when pruned using SparseGPT. Interestingly, OPT350m with Wanda pruning yields the lowest BertScore among all models, while OPT350m with SparseGPT pruning achieves the highest. The strong performance of OPT125m with Structured 4:8 pruning likely stems from its ability to prune neurons within a larger window, enabling access to more information during decision-making.

A similar trend is observed with the ROUGE score, where OPT350m with SparseGPT pruning achieves the highest score, and OPT350m with Wanda pruning performs the worst.

However, as we will see shortly, both BertScore and ROUGE fail to adequately capture the "global coherence" of the outputs—an aspect that is somewhat independent of metrics like keyword-phrase matching or topical/contextual similarity measured by these scores. This necessitates a qualitative analysis of the models' performance.

4.3 Qualitative Error Analysis

Looking first at the OPT125m models that were not pruned, we see that fine-tuning the model leads to substantial improvement in performance over the base (0-shot) model, with respect to ROUGE and BertScore metrics. This is what one would expect from a fine-tuned model, as compared with 0-shot. Notably, the base and full models are much faster during inference than the QLoRa fine-tuned model, despite the latter technically having more parameters. To complicate things further, they take up exactly twice the disk space. The reason for this is that the QLoRa model is quantized, whereas the other two are not. The double-precision of the base and full models causes them to take twice the space on disk. The reason these models are nevertheless faster has to do with the fact that PyTorch, the training framework we used, is optimized for inference with full-precision models, leading to significant speed-ups. The additional parameters in the QLoRa model are a result of the way that QLoRa adds parameters to the model rather than adjusting or deleting pre-existing ones, but their influence likely contributed minimally to the longer inference time, in comparison to the slow-down incurred from quantization. The same basic trend is present amongst the OPT350m models, with the notable caveat that the Base (0-shot) OPT350m model performed miserably in comparison to even the 125m models, according to ROUGE and BertScore.

Looking now at the pruned OPT125m models, we observe that Wanda applied with 50% sparsity, structured 2:4 pruning, and structured 4:8 pruning led to almost no observable difference in performance, from the perspective of ROUGE and BertScore. This confirms the idea that pruning connections based on activation

leads to the removal of non-essential neural network components, in turn providing evidence for the Hypotheses that a highly-specialized sub-network is responsible for most of the performance in a trained large language model. Notably, Wanda outperformed other approaches such as Magnitude pruning and Sparse-GPT, confirming prior hypotheses about the importance of activation, and not just magnitude, for pruning. Again, the same basic pattern holds amongst the 350 models we tested, which saw only a minor degradation in performance.

Unfortunately, the use of pruning, whether structured or not, did not have a significant impact on the number of parameters, inference time, or disk size of the model. This is surprising, since we expected structured pruning to remove entire channels of the weights matrices. Upon further review, common implementations of Wanda set pruned weights to zero but does not follow up by removing columns or rows that are all zero. This makes some sense if we consider that structured N:M pruning refers to pruning N weights for every M contiguous weights. If this happens along the rows, then we cannot expect a row that is not of length M to be fully pruned. As the number of columns increases, we can also expect the probability of a prunable “stack" of zeros showing up as an unintended consequence of row-wise pruning to approach zero, making column-wise pruning impossible. This result sheds light on the difficulty of pruning in such a way as to actually make weights matrices denser and easier to parallelize on GPU’s. It also suggests “structured N:M pruning" is not actually “strcutured pruning" in the traditional sense of that term, indicating a poor choice of naming on behalf of the former.

Qualitatively, the OPT-125m QLoRA fine-tuned Wanda 2:4, Wanda 4:8, and Wanda unstructured models all perform poorly on the validation set. Their generated outputs merely repeat the beginning section of the input text. Wanda unstructured is slightly better in that it sometimes adds a short phrase indicating the bill that was amended, as opposed to a verbatim repeat from the start. For instance, in example 2, it starts with "The Victims’ Bill of Rights Act of 2008 (Marsy’s Law) is amended to read:" as opposed to "The people of the State of California do enact as follows: SECTION 2. Section 1170.02 is added to the Penal Code, to read:". For example 3 from the test set, OPT-125m Wanda 2:4 again repeats verbatim the beginning section of the input text. But Wanda 4:8 and Wanda unstructured are able to provide a summary.

Wanda 4:8:

"Special Agent Scott K. Carey Public Safety Officer Benefits Enhancement Act - Amends the Special Agent Scott K. Carey Public Safety Officer Benefits Enhancement Act to include a public safety officer who is eligible to receive benefits under this subpart if such officer dies in the line of duty on or after January 1, 1978; an"

Wanda unstructured:

"Special Agent Scott K. Carey Public Safety Officer Benefits Enhancement Act - Amends the Omnibus Crime Control and Safe Streets Act of 1968 to include an eligible dependent of a public safety officer who is eligible to receive benefits under this subpart if such officer dies in the line of duty on or after January 1, 1978; and"

While the wording seemed similar, the Wanda unstructured output is actually better because it correctly mentions the dependent of the officer receiving the benefits instead of the officer receiving it.

The OPT-125m QLoRA fine-tuned magnitude 2:4 and SparseGPT 2:4 models do not generate coherent output summaries. For the validation set, magnitude 2:4 generates blanks while SparseGPT 2:4 only generates a header such as "State of California Criminal Justice Act of 2008 (Marsy's Law) - Amends the Penal Code to read:". For example 3 from the test set, magnitude 2:4 repeats verbatim a section at the end of the input text while SparseGPT 2:4 repeats verbatim the beginning of the input text.

The OPT-350m QLoRA fine-tuned Wanda 2:4 model generates novel text for summaries but which are often irrelevant to the input text. Consider example 2:

"(a) In this section, the term "prisoner" means a person convicted of first-degree murder, a person sentenced to death or life in prison without possibility of parole, a person sentenced to life or death in prison without possibility of parole, a person sentenced to death or death in prison without possibility of parole, a person sentenced to death or death in prison without possibility of parole, a person sentenced to death or death in prison without possibility of parole, a person sentenced to death or death in prison without possibility of parole, a person sentenced to death or death in"

The output uses a lot of the same key words like "prisoner", "first-degree murder", "parole" but it's trying to define the term "prisoner", which was not the point of the bill.

In example 1, it generates a good summary:

"The Transportation and Intercity Rail Capital Program is hereby created to fund transformative capital capital improvements, as defined in subdivision (d), that will modernize California's intercity, commuter, and urban rail systems and bus and ferry transit systems to achieve all of the following policy objectives: (1) Reduce emissions of greenhouse gases. (2) Expand and improve transit service to increase ridership. (3) Integrate the rail service of the state's various rail systems."

For the test set, it surprisingly does not generate coherent summaries. Consider example 3 output:

"SEC. 1. SHORT TITLE. This section is called the "Special Agent Scott K. Carey Advisory and Training Assistance Act". SEC. 2. SUMMARY. The term "Special Agent Scott K. Carey" means the Director of the Office of Special Counsel."

The OPT-350m QLoRA fine-tuned magnitude 2:4 model also generates blanks

for the validation set examples. It does summarize example 3 from the test set as follows: "Authorizes the Secretary of Justice to revise the definition of ‘eligible dependent’ in this section." While the summary is incorrect, it at least attempts to summarize and represents an improvement from the OPT-125m magnitude 2:4 model.

The OPT 350m QLoRA fine-tuned sparsegpt 2:4 model generates decent summaries for both validation and test sets.

For example 2 (validation set), it generates the following:

"Amends the Victims’ Bill of Rights Act of 2008 to make it clear that a prisoner sentenced to death or life in prison without possibility of parole is eligible for medical parole if the prisoner is permanently medically incapacitated with a medical condition that renders him or her permanently unable to perform activities of basic daily living. Amends the Sentencing Reform Act of 1996 to make it clear that a prisoner sentenced to death or life in prison without possibility of parole is eligible for medical parole if the prisoner is permanently medically incapacitated with a medical condition that renders him or her permanently unable to perform activities of basic daily living."

For example 3 (test set), it generates the following:

"Amends the Omnibus Crime Control and Safe Streets Act of 1968 to provide for the replacement of the "durable" and "automobile" benefits for public safety officers who are disabled. Amends the Children’s Health Insurance Program (CHIP) to provide for the replacement of the "child" and "wife" benefits for public safety officers who are disabled. Amends the Children’s Defense Fund Act of 1996 to provide for the replacement of the "child" and "wife" benefits for public safety officers who are disabled. Amends the Children’s Defense Fund Act of 1996 to provide for the"

The generated summaries roughly reflect the content of the input text, but they suffer from repetition.

In summary, from a qualitative point of view, the rank order of the different pruned models from best to worst is as follows: OPT-350m sparsegpt 2:4 > OPT-125m Wanda unstructured > OPT-125m Wanda 4:8 > OPT-350m Wanda 2:4 > all other models (they generate blanks or merely repeat the input text). Larger models do generally lead to better performance, as evidenced by the higher quality output of sparsegpt 2:4 and Wanda 2:4 going from 125m to 350m. It is also unsurprising that being the worst pruning method, magnitude 2:4 see little increase in quality as the model size increases. We also see that among the Wanda sparsity types, Wanda unstructured gives the best results, followed by Wanda 4:8. Wanda unstructured provided the best summary due to more freedom in selecting the best weights to keep, while 4:8 outperforms 2:4, suggesting that contiguous weights are important in the feature representation of the text. Sparsegpt and Wanda have similar output

qualities and are better than magnitude, all else being equal. Fully fine-tuned models are much better than QLoRA fine-tuned models. Even OPT-125m fully fine-tuned model outperforms the OPT-350m QLoRA fine-tuned model, suggesting that at small model sizes, fine tuning is critical to quality model output.

5 Conclusions

In conclusion, our study highlights several critical nuances in model pruning and optimization techniques. We demonstrate that structured N:M pruning diverges from traditional structured pruning by not effectively removing weights, which limits its performance advantages over unstructured pruning. From a performance perspective, unstructured pruning remains preferable, offering greater flexibility and often better retention of model accuracy. Additionally, quantization significantly enhances runtime efficiency, though it requires careful implementation to balance speed gains with potential accuracy loss.

When comparing pruning methods, activation-based pruning does not consistently outperform magnitude-based pruning unless the added cognitive complexity justifies its use. This underscores the importance of selecting pruning strategies based on specific model and task requirements. Notably, smaller models are particularly vulnerable to performance degradation from structured pruning, making fine-tuning essential to preserve their effectiveness.

Our findings also reveal that models can artificially inflate metrics like ROUGE and BERTScore by repeating text segments, which calls for more robust evaluation methods. Furthermore, small decoder-only models exhibit a tendency to overfit when handling tasks with long prompts and short outputs, hindering their ability to generalize effectively. In this context, QLoRa fine-tuning appears insufficient for optimizing pruned small decoder-only models for summarization tasks, indicating a need for more advanced fine-tuning approaches.

Overall, these insights emphasize the necessity for a nuanced and context-aware application of pruning, quantization, and fine-tuning techniques. Balancing efficiency with performance requires a deep understanding of each method's strengths and limitations, particularly when working with smaller models or specific task requirements. By carefully considering these factors, practitioners can enhance model performance and efficiency in real-world applications.

6 Acknowledgements

References

[Dassum(2024)] Dassum. 2024. https://dassum.medium.com/fine-tune-large-language-model-llm-on-a-custom-dataset-with-qlora-fb60abdeba07 Fine-tune large language model (llm) on a custom dataset with qlora. Accessed: 2024-11-29.

[Eidelman(2019)] Vladimir Eidelman. 2019. https://doi.org/10.18653/v1/d19-5406 Billsum: A corpus for automatic summarization of us legislation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, page 48–56. Association for Computational Linguistics.

[Frantar and Alistarh(2023)] Elias Frantar and Dan Alistarh. 2023. http://arxiv.org/abs/2301.00774 Sparsegpt: Massive language models can be accurately pruned in one-shot.

[Han et al.(2015)Han, Pool, Tran, and Dally] Song Han, Jeff Pool, John Tran, and William J. Dally. 2015. http://arxiv.org/abs/1506.02626 Learning both weights and connections for efficient neural networks.

[Hu et al.(2021)Hu, Shen, Wallis, Allen-Zhu, Li, Wang, Wang, and Chen] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. http://arxiv.org/abs/2106.09685 Lora: Low-rank adaptation of large language models.

[HuggingFace(2024)] HuggingFace. 2024. https://huggingface.co/docs/transformers/en/model$_doc/optOptmode$
$2024 - 11 - 01$.

[Lin(2004)] Chin-Yew Lin. 2004. https://aclanthology.org/W04-1013 ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

[Sun et al.(2024)Sun, Liu, Bair, and Kolter] Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2024. http://arxiv.org/abs/2306.11695 A simple and effective pruning approach for large language models.

[Zhang et al.(2020)Zhang, Kishore, Wu, Weinberger, and Artzi] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. http://arxiv.org/abs/1904.09675 Bertscore: Evaluating text generation with bert.