Journal of Computer Science

Original Research Paper

A Conceptual Framework for Efficient and Sustainable Pruning Techniques in Deep Learning Models

Nyalalani Smarts, Rajalakshmi Selvaraj and Venumadhav Kuthadi

Department of Computing and Informatics, School of Pure and Applied of Sciences, Botswana International University of Science and Technology, Palapye, Botswana

Article history
Received: 14-10-2024
Revised: 13-12-2024
Accepted: 23-12-2024

Corresponding Author:
Nyalalani Smarts
Department of Computing and Informatics, School of Pure and Applied of Sciences, Botswana International University of Science and Technology, Palapye, Botswana
Email: nyalalani.smarts@studentmail.biust.ac.bw

Abstract: This research paper proposes a conceptual framework and optimization algorithm for pruning techniques in deep learning models, its focus is on key challenges such as model size, computational efficiency, inference speed and sustainable technology development. The aim of the framework is to transition from large neural networks to sparse, efficient models, indicating the benefits of pruning in improving model scalability and applicability of the pruned models. The proposed framework focuses on reducing the model size, optimizing training schedules and facilitating efficient deployment in real-world devices. The development of the framework involves four stages: Reviewing critical research concepts, identifying relationships between concepts and designing the pruning framework. Furthermore, this study also introduces a new multi-objective optimization algorithm that formalizes the trade-offs between accuracy, computational cost, inference time and energy consumption in the pruning process. Our experiments demonstrate the method's effectiveness in achieving notable model compression while preserving competitive performance on a sentiment analysis and linguistic acceptability tasks using Stanford Sentiment Treebank (SST-2) and Corpus of Linguistic Acceptability (CoLA) datasets. The results of our experiments show the BERT Base model being pruned to 25 million parameters gaining an accuracy of 96.3% on SST-2 dataset and F1-score of 95.2%. Furthermore, the pruned model demonstrated F1 score of 82.3 and 56% of Matthews correlation coefficient in CoLA dataset respectively. This framework, along with the algorithm, serves as a reference for researchers and practitioners, who can select a suitable approach based on the specific application requirements for pruning deep learning models.

Keywords: Deep Language Models, Pruning, Efficiency, Pretrained Language Models, Sustainable Technology Development

Introduction

Natural Language Processing (NLP) domain has experienced significant progress with the growth of large language models based on transformer architecture. These models such as Bidirectional Encoder Representation from the Transformers (BERT) by Devlin *et al*. (2019) and Generative Pre-trained Transformer (GPT) reported in a study by Yenduri *et al*. (2024) have shown outstanding performance on a wide range of NLP tasks. The models have excelled in tasks such as classification, machine translation and question-answering as revealed by Elazar *et al*. (2021). However, the increasing size and complexity of the models (Khan *et al*., 2019), having billions of parameters, requires a significant number of computational resources for training and inference. As a result, rising energy and performance costs have sparked interest in reducing the size of neural networks through techniques such as selective pruning as stated by Cheng *et al*. (2024). Marinó *et al*. (2023) in their study suggested that this has led to researchers starting to have more interest in pruning techniques which can selectively reduce model size and complexity, therefore offer potential solutions to lower energy consumption and computational demands. These make large models more accessible for deployment in resource-constrained environments like in mobile devices (Lu and Lyu, 2021; Cai *et al*., 2022). The studies by Cheng *et al*. (2024);

Wang *et al*. (2024) have also shown that the process of pruning presents a great possible solution to make NLP research more accessible and sustainable, thereby supportive of study by Marinó *et al*. (2023).

Studies have shown that available pruning techniques and frameworks for pre-trained language models have their own shortfalls. The said shortfalls can be attributed to factors such as computational overhead, loss of accuracy or performance, difficulty in determining optimal pruning criteria, or lack of consideration for sustainable technological development. According to the study of Chitty-Venkata *et al*. (2023) it has been discovered that computational overhead creates a sizable hurdle to model adoption, because substantial time and resources are needed for pruning large language models. For example, the study by Zhu *et al*. (2023) have suggested that loss of accuracy can happen because of removal of important weights during pruning, which can affect downstream task performance. Deciding on the best pruning criteria depends on the specific model at hand, the task to be performed and target sparsity level, which adds complications to the pruning process (Yang *et al*., 2022; Yeom *et al*., 2021). Additionally, in related research Chitty-Venkata *et al*. (2023), have also noted that some of the pruning methods fail to factor type and sustainability of pruning results in unnecessary environmental harm. The cited studies highlight that current pruning techniques suffer from key limitations, including computational and environmental concerns. Therefore, it is crucial to advance the knowledge about these challenges and gaps for improving the efficacy and efficiency of pruning in developing merely effective and sustainable language models. As highlighted earlier, key barriers including computational overhead, reduced model accuracy, the complexity of defining optimal pruning criteria, and inadequate attention to sustainable development impede the broader adoption of pruning techniques in practice.

Therefore, in this study we introduce a conceptual framework and a novel multi-objective optimization algorithm that formalizes pruning as a trade-off between accuracy, computational cost, inference speed and energy consumption. Through an analysis of these conceptual relationships, along with resource efficiency and energy consumption, the proposed framework and multi-objective optimization algorithm seek to establish practical guidelines for enhancing computational resource utilization and minimizing energy consumption in NLP applications via structured pruning techniques. This study aims to look for a balance between the size of the models and their accuracy and, also to provide green directions in Natural Language Processing. This study contributes the following:

- A conceptual framework for guiding pruning techniques, schedules, deployment considerations and model optimization
- Survey of key research concepts including model pruning, efficiency, inference and sustainable technology development
- Identifying connections between key research concepts
- Design and development of the proposed conceptual framework for pruning techniques
- A novel multi-objective optimization algorithm
- Evaluation of the pruned model against benchmarks

Related Works

Pruning techniques in deep learning represent yet another key method of downscaling the size of neural network models while at the same time enhancing their accuracy. These techniques can be broadly categorized into two main branches: Structured pruning and unstructured pruning, each method offering distinct approaches to addressing the challenge of overparameterization.

Unstructured Pruning

Unstructured pruning, defined by Gupta and Agrawal (2022), as identifying and eliminating less important individual weights from a neural network. The concept of unstructured pruning received its initial groundwork from pioneers Poppi and Massart (1998); Hassibi *et al*. (2002) who used magnitude weight pruning. The approach first identifies weights of little significance to the model outcomes before removing them hence achieving an accurate reduction of the size of the model without necessarily reducing its efficiency. The understanding of pruning methods helps researchers create solutions that enhance both performance quality and model size reduction without altering accuracy rates.

The need to optimize the number of parameters in deep learning models for efficient consumption of resources by constrained devices is one the main reasons behind pruning techniques. The technique works, by making some connection values as zero or deleting them out reduces the weight value and hence, makes a model more resource friendly. In Wiedemann *et al*. (2020), it was observed that using a method of trimming the lower weight connections, greatly decreased the size of the model (Liang and Liu, 2015). In addition, the magnitude-based pruning method offers a graceful trade-off between size and performance, as according to Gerum *et al*. (2020). While unstructured pruning may have some challenges when it comes to sparsifying matrices on some frameworks or hardware, this method has one universal benefit, it can be used with any network architecture. The pruning leads to the fact that deep learning models may be efficiently optimized for further deployment to different classes of devices, thus promoting accessibility and performance in the constrained settings.

Structured Pruning

Neural network compression exclusively benefits from the structured pruning methodology which constitutes its own subspace within pruning methods. The structured approach gradually deletes entire weight structures or blocks within the weight networks of large pre-trained neural networks. An *et al*. (2024) confirm together with Wang *et al.* (2020) that structured pruning methods operate across structures or blocks of weights. Structured pruning achieves significant reductions of model size and inference expenses as per the documented research findings. Structured pruning enables modifications to blocks of weights along with attention heads and individual layers or entire structural parts of a neural network according to Li *et al*. (2020); Zhang *et al*. (2021); Shim *et al*. (2021) respectively. This technique provides excellent opportunities for compression along with cost reduction by enabling the removal of significant weight blocks compared to the base approach.

Knowledge advancement for professionals comes from intense training in structured pruning approaches which enable them to optimize model dimensions and computing speed during neural network development. Research teams can develop efficient real-world high-performance models for various deployment scenarios through combining structured pruning techniques with other compression and pruning methods according to Cai *et al*. (2022).

Limitations and Gaps in Current Prunning Approaches

The available pruning methods and frameworks for deep language models have shortcomings and with some questions left unanswered. These deficiencies stem from multiple factors including computational inefficiency, accuracy degradation, challenges in establishing optimal pruning criteria, and insufficient consideration of sustainable technology development principles. Pruning large language models is extremely computationally expensive; it takes a large amount of time and resources, especially during training (Xiao *et al*., 2024). Such degradation in accuracy might arise from pruning some important weights, which impacts the performance of downstream task (Cheng *et al*., 2024). As for the choice of, pruning criteria are dependent on the model, task and desired sparsity level and is thereby a delicate process (Yeom *et al*., 2021; Li *et al*., 2024). Moreover, there are other pruning strategies that failed to address sustainability issues; many of the pruning methods cause unnecessary harm to environment (Xiao *et al*., 2024). These studies demonstrate that contemporary pruning methodologies encounter multifaceted challenges spanning computational efficiency to ecological sustainability considerations. Strengthening these limitations and gaps this project is the critical step forward in addressing the future developments of more efficient and sustainable pruning techniques in order to unleash the full potential of deep neural network models across various sectors, their applications while avoiding negative environmental impacts.

Materials and Methods

This research presents the new sparsification-driven framework in an effort to develop optimized models for sustainable technological development. The proposed framework seeks to leverage sparsity principles in deep learning to facilitate the deployment of compressed models onto resource-constrained devices, aligning with sustainable long-term technological development objectives. Efficient models are those that optimize footprint metrics such as model size, inference latency and training time while minimizing quality loss and improving model generalization. Sparsity in deep learning enables compression techniques to target the representational efficiency of over-parameterized models (Menghani, 2023). “Sustainable technological development involves articulating functions to meet future demands and societal needs while reducing environmental impact” (Vergragt and Jansen, 1993).

However, sparse-centric approaches to sustainable technological development are currently lacking. The framework aims to make neural networks more efficient, sustainable and effective in real-world applications by illustrating techniques for reducing their size, optimizing their training schedules and considering practical deployment factors.

The following steps were taken to develop the framework:

1) An analysis of literature on the fundamental ideas pertaining to the key research topics that include model pruning, efficiency, inference and sustainable technology development
2) identifying connections between key research concepts
3) design and development of the proposed conceptual framework for pruning

Reviewing Literature on Specific Research Concepts

This study has undertaken a rigorous and structured approach to determine the relationships between the key concepts of this research. The structured approach is based on Kitchenham guidelines (Kitchenham and Brereton, 2013), which include the preparation of research questions, a search strategy and research selection criteria.

Planning the Review

The general research question which was identified is as follows: How can model pruning be used to enhance efficiency and support sustainable technological development in deep learning models? To make it possible to further explore the problem, the general question was divided into three narrower sub questions:

1. In what way does pruning help make deep learning models more efficient?
2. What is the association between model efficiency and inference performance in sparse models?
3. How exactly does increased model efficiency impact the further development of sustainable technologies?

Developing a Review Protocol

The objective is to create a conceptual framework that guides the implementation of pruning techniques to improve the efficiency and sustainability of deep learning models. For the search strategy in this study, the key concepts represented the search terms that were applied to a variety of resources. Boolean search strategy is defined as the process of converting a research question into a research string with the view of querying a database or search engine, (Sivarajkumar *et al*., 2024). The academic internal databases employed in this study include IEEE Xplore, Google Scholar, PubMed, semantic scholar and ACM Digital Library. Keywords are "model pruning," "deep learning efficiency," "neural network optimization" and "sustainable AI.

To fulfill the inclusion criteria, we restricted our sources to the articles from the referred scientific journals, evaluation reports and technical papers and conference papers of peer review. More precisely, we only considered papers that focused on the effects of pruning on the model as well as papers that focused on the broader issue of sustainability in AI and deep learning. We prioritized articles that demonstrated a link between the search terms and described the relationships between them. Additionally, we included articles addressing the search terms in conjunction with any of the four core concepts of this study: Model pruning, efficiency, inference and sustainable technology development (Shao and Zhang, 2020).

The articles were excluded based upon the following: Articles that did not relate directly to deep learning or pruning techniques, papers and articles with no prior experimental results or data, the papers with the search terms but not looking into the connection or relevance between them were considered irrelevant and thereby excluded. We created a standardized data extraction form to capture relevant information from each study to achieve the data extraction methods, including pruning methods, metrics for efficiency, impacts on inference and sustainability considerations.

Conducting the Review

The identification of research articles was achieved by using the ‘AND’ and ‘OR’ Boolean operators to construct search strings to extract articles that join terms based on research questions. The search strings included the following "model pruning AND deep learning efficiency," "pruning techniques AND inference performance" and "sustainable AI AND neural networks." The strings for the OR operator included "model pruning OR deep learning efficiency," "pruning techniques OR inference performance" and "sustainable AI OR neural networks."

In the process of selecting primary studies, initially, we searched and selected studies based on the titles and the abstracts of the studies and then, we checked and selected the full text of these studies. Of 500 original identified articles, 120 papers were reviewed after excluding irrelevant and low-quality articles while 46 were reviewed in this final review. Data was systematically extracted using the predefined form and the extraction process was monitored to maintain consistency and accuracy. Some of the data considered during extraction involved the type of pruning technique applied whether structured or unstructured, efficiency measurements such as model size and inference latency and whether sustainability impacts have been realized such as energy efficient consumption.

Identification of Relationships Between Key Concepts

These interconnections were established from a systematic review of the literature in the field. To identify studies that reported on the effects of pruning on the performance of neural network models, we concentrated on papers that described pruning approaches and their results. We analyzed and discussed relevant literature concerning sustainability in AI and deep learning. Any articles that showed utilization of the search terms and explained the correlation between the identified search terms were included in the study. This involved reviewing different articles to see how model pruning affects efficiency and inference and how such gains fit sustainable technology progress (Singh and Gill, 2023). We began by analyzing the effect of model pruning in model efficiency. For instance, Tay *et al*. (2023); Cho *et al*. (2021) suggested that pruning led to smaller model sizes and shorter inference time as well as momentous acceleration of the training phase. Menghani (2023) has further supplemented these findings indicating the top goal of pruning as being to achieve different footprint goals.

Subsequently, we examined the correlation between model pruning and inference. Several studies Baccour *et al*. (2024); Ebrahimi *et al*. (2023); Abdi *et al*. (2023) have demonstrated that only a subset of network parameters significantly contributes to model performance. Based on

this observation, they concluded that strategic pruning can be implemented with minimal impact on model accuracy. In the study of He and Xiao (2024), different forms of structural pruning approaches were described and the work of Jin *et al*. (2024) pointed out that properly integrated approaches were required.

Last but not least, we assessed the link between efficiency and the development of sustainable technologies. Schwartz *et al*. (2020) proposed the notion of Green AI and focused on efficiency as well as environmental preservation Bolón-Canedo *et al*. (2024); van Wynsberghe (2021), Salehi and Schmeink (2024). Adding to the previous studies Zhang *et al*. (2021); Huang *et al*. (2022); Strubell *et al*. (2020) presented the measures to compare the efficiency of AI models and environmental impact.

Design and Development of the Pruning Framework

The relationships identified in the section identification of relationships between key concepts above were used to design and develop the conceptual framework. The output of the identified relationships is the proposed framework in Fig. (1). The proposed framework illustrates a series of steps that shows model pruning pathway to show the efficiency of compressed models on edge devices in order to achieve sustainable technological development goals.

Datasets and Preprocessing

Our approach includes the utilization of a well-established public dataset which comprises of the SST-2 dataset as reported by (Wankhade *et al*., 2022) and CoLA dataset as reported by (Warstadt *et al*., 2019) with training and development sets, from the General Language Understanding Evaluation (GLUE) benchmark. In our experimental design, we employed random seed initialization to shuffle and reorder the training datasets, thereby ensuring varied data exposure sequences during model training. The pre-trained BERT model checkpoint, BERT-base-uncased and BERT-large were employed as the foundational models.

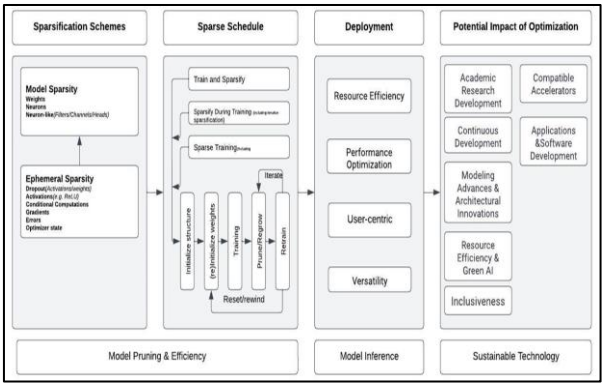

Fig. 1: Proposed conceptual framework

The SST-2 dataset is a binary sentiment classification for positive/negative sentiments used for sentence-level sentiment analysis tasks. Its standardized labels, well-formed sentences and accessibility make it highly suitable for NLP experiments. We also realized that further preprocessing was not necessary because the dataset has been extensively validated in NLP research, featuring properly structured textual data and consistent labeling conventions. It is comprised of 67,000 training samples and 872 development samples, which we evaluated based on accuracy and F1 score.

Likewise, the CoLA dataset which is designed for linguistic acceptability classification, was employed to evaluate the model's understanding of grammatical correctness. It contains labeled sentences which are categorized as either acceptable or unacceptable and were curated by linguists to ensure high data quality and reliability. We finetuned our model with CoLA dataset containing a training set of 10,000 samples and a development set of 1,000 samples, also evaluated using accuracy, F1 score metrics and Matthews Correlation Coefficient (MCC).

Training Configurations

We selected precise training parameters for process optimization then set the learning rate at 1e-5 along with a batch size of 16 and executed 5 iterations. The training arguments included parameters for saving the model, loading the best model at the end of training, setting the number of training epochs to three and defining the evaluation and saving strategies. Accuracy tests and F1 score evaluations were used to identify the best model among the pre-defined 10 saved model variants. To ensure the model training efficiency and robustness at the same time preventing overfitting while maximizing performance the configurations were designed as mentioned above.

Optimization Methods of the Pruning Process

There exist several algorithms which have been developed to optimize the pruning process, concentrating on balancing model size reduction while maintaining performance. The optimization methods in general are important for boosting the performance and efficiency of deep learning models, especially Pre-trained Large Language Models (PLMs) like BERT and GPT etc. The study by Michel *et al*. (2019) which have previously used the Attention Head Pruning proved that pruning attention heads in transformer models can lead to smallest degradation in performance while significantly reducing the model size. It was observed that a maximum of 40% attention heads could be cut off BERT without a noticeable drop in accuracy. The method called Block Structural Pruning which was introduced by Lagunas *et al*. (2021), eliminates attention-heads structured and paired rows/columns in feed-forward layers. This technique

resulted in a 2.4× faster inference with only a slight decrease in predictive performance.

The movement pruning method proposed by Sanh *et al*. (2020) which helps to achieve improved performance in high sparsity regimes by concentrating on the weights that are changing during training. This approach has been successfully implemented to BERT and it has led to enhancements in improved accuracy as compared to traditional magnitude pruning (Gupta and Agrawal, 2022). The DistilBERT technique outlined in Sanh *et al*. (2019) is a less complex version of BERT attained through knowledge distillation. This model retains 97% of BERT's language understanding capabilities while being 60% smaller and faster (Tay *et al*., 2023). Instead of selecting a single fixed pruning scheme, adaptive pruning techniques adjust the pruning strategy based on the model's performance and the importance of different layers or components. The study by Yao *et al*. (2021) proposed LEAP, which uses learnable pruning ratios that are adjusted during training. This method allows for different pruning levels across layers, optimizing the pruning process without extensive hyperparameter tuning. One-shot pruning techniques prune the model in a single step using the importance scores, then standard training is done to fine-tune the sparse model. The one-shot pruning method introduced by Kurtic *et al*. (2022) proved that it can significantly outperform gradual pruning methods, achieving high sparsity without the need for extensive fine-tuning. Pruning methods can also target optimization of energy consumption and enhancing inference speed which are critical for deploying models in resource-constrained environments (Hasan and Alam, 2023).

Our proposed novel optimization method offers a methodical way to pruning deep learning models, concentrating on optimizing performance, computational cost, inference time and energy consumption. The process starts by initializing the weights W of the original model, including the model's accuracy inputs $A(W)$, acceptable accuracy loss threshold ϵ, computational cost $C(W)$, inference time $T(W)$, energy consumption $E(W)$ and an optional target pruning ratio p. During the pruning phase, a target pruning ratio p is specified and $p \cdot 100\%$ of the less essential weights, using their magnitude to identify them, pruned weights W are created through removal of those weights. Computational cost $C(W')$ of the pruned model is then assessed to confirm substantial reductions while maintaining acceptable accuracy $A(W')$, where $A(W') \geq A(W) - \epsilon$. The following step deals with inference efficiency, where time taken is recorded as $T(W')$, focusing on reductions while keeping in view the accuracy thresholds. After this, the sustainability of the pruned model is examined by calculating its energy consumption $E(W')$ and minimizing it under the same level of accuracy. In some cases where the pruning ratio obtained does not give satisfactory performance, the algorithm iteratively adjusts p to balance pruning efficiency and performance. The final multi-objective optimization step simultaneously minimizes $C(W')$, $T(W')$ and $E(W')$ while making sure that accuracy remains above the threshold. Finally when the objectives shows that they have been met, the algorithm outputs the optimized pruned model W' along with its performance metrics. This method ensures a streamlined approach to model compression, effectively balancing accuracy, efficiency and sustainability.

Pruning Selection Criteria

The importance of pruning selection criteria is to identify which weights or parameters to remove from a model in order to gain the wished sparsity while maintaining performance. In this study the Magnitude-Based Pruning (MBP) approach was adopted. The strategy eliminates weights with small magnitudes because these components are considered to have minimal impact on model predictions. In the study done by Gupta and Agrawal (2022) they proved that the approach could significantly reduce model size with minimal accuracy loss. Hyperparameter tuning (Saleem *et al*., 2024) is significant while working with optimizing the performance of the pruning algorithm, involving the adjustment of parameters that control the pruning process, like the target pruning ratio p and the acceptable accuracy loss threshold ε. The training effectiveness of pruning is influenced by using adaptive learning rates approach together with Adam optimizer for rate adjustments. The process is illustrated in detail in the following steps below.

Step 1: Model Pruning

I. Initialization: Start with the full model weights W
   - A BERT-base model with 110M parameters

II. Set target pruning ratio p: Select the percentage of weights to prune, $0 \leq p \leq 1$
   - Target pruning ratio $p = 0.3$, which means pruning 30% of the weights

III. Prune weights: Identify and remove $p \cdot 100\%$ of the least important weights, e.g., those with the smallest magnitudes
   - If $W = [0.8, -0.5, 0.03, -0.1]$ and $p = 0.5$, prune the smallest two weights $(0.03, -0.1)$, resulting in $W' = [0.8, -0.5]$

The reason behind the MBP approach is that small-magnitude weights contribute little to the model's output and can be removed without affecting much on accuracy. It can be represented as $= \frac{|W - W'|}{|W|}$, where W represents the original weights and W' the pruned weights. This approach is proved by works like that of (Han *et al*., 2015) which indicated that pruning small-magnitude weights can entirely reduce model size while maintaining performance.

Step 2: Model Efficiency

I. Calculate computational cost $C(W')$: Determine the FLOPs of the pruned model. The goal is to reduce FLOPs compared to the original model.
   - A pruned BERT model may reduce FLOPs from 10-2 billion

II. Verify accuracy: We ensure that the pruned model's accuracy $A(W')$ meets $A(W') \geq A(W) - \epsilon$
   - If the original accuracy is 90% and $\epsilon =$ 1%, the pruned model's accuracy must remain at or above 89%

The process of Accuracy Preservation Constraint involves verifying that the pruned model's accuracy, $A(W')$, satisfies $A(W') \geq A(W) - \epsilon$, where ϵ is the acceptable accuracy loss threshold. This constraint guarantees the pruned model keeps performance close to the original, matching efficiency improvements with accuracy preservation.

Step 3: Inference Efficiency

I. Measure inference time $T(W')$: We evaluate the time required to process an input. The objective is to reduce this time while maintaining acceptable accuracy
   - A baseline BERT model takes 50ms per input. After pruning, the inference time drops to 30ms.

Step 4: Sustainability

I. Compute energy consumption $E(W')$: Using NVIDIA power profiler to measure energy usage. Minimize energy consumption while ensuring accuracy thresholds
   - If the original energy consumption is 100 Joules, pruning might reduce it to 70 Joules

Sustainability Metrics focus on assessing energy consumption $E(W')$ and inference time $T(W')$ to evaluate the environmental and operational impact of pruning. NVIDIA Management Library (NVML) for GPUs and Intel Power Gadget for CPUs were used to measure these metrics. The point is to ease hardware and energy demands, guaranteeing the pruned model remains efficient and environmentally friendly while maintaining acceptable performance levels.

Step 5: Optimal Pruning Criteria

I. Optimize pruning ratio p: If the initial pruning ratio leads to unacceptable performance losses, iteratively adjust p to achieve an optimal balance
   - Start with $p = 0.3$ If $A(W')$ drops below $A(W) - \epsilon$, reduce p to 0.2 and re-evaluate

Iterative Adjustment of Pruning Ratio aims to enhance the pruning ratio p in sequential stages to balance pruning success and performance maintenance. The objective is to maximize the extent of pruning while ensuring the pruned model satisfies $A(W') \geq A(W) - \epsilon$. This approach follows the methodology proposed by Frankle and Carbin (2018) in their IMP framework, employing iterative pruning and evaluation cycles to identify optimally sparse, high-performance subnetworks.

I. Simultaneous optimization: minimize $C(W')$, $T(W')$ and $E(W')$ while ensuring $A(W') \geq A(W) - \epsilon$

For a trade-off scenario:

- Option 1: FLOPs = 35B, accuracy = 89.5%, inference time = 32ms, energy = 75J
- Option 2: FLOPs = 33B, accuracy = 89%, inference time = 30ms, energy = 70J
- Select the configuration offering the best balance based on deployment needs

This approach ensures that pruning improves efficiency without overly sacrificing speed or introducing significant accuracy loss. It aligns with methods discussed by Yeom *et al*. (2021) which emphasize the importance of balancing accuracy, speed and efficiency for practical, real-world applications.

Final Output

Return pruned model W': Once the optimal trade-offs are achieved, output the final pruned model W', along with metrics like $A(W')$, $C(W')$, $T(W')$ and $E(W')$.

Algorithm 1: Efficient model pruning optimization algorithm

Input :
- Original model weights W
- Accuracy of the original model $A(W)$
- Acceptable accuracy loss threshold ϵ
- Original computational cost $C(W)$
- Original inference time $T(W)$
- Original energy consumption $E(W)$
- Target pruning ratio p (optional, can be tuned)

Output :
- Pruned model weights $'W'$
- Pruned model's accuracy $A(W')$
- Optimized computational cost $C(W')$
- Optimized inference time $T(W')$
- Optimized energy consumption $E(W')$

Step 1 **Model Pruning**

I **Initialize:** Start with the original weights W

II **Set target pruning ratio:** Choose a target pruning ratio p, where,
$0 \leq p \leq 1$

III **Prune weights:**
- Compute the pruned weights W' by removing $p \cdot 100\%$ of the least important weights from W
- $p = \frac{|\mathbf{W}-\mathbf{W}'|}{|\mathbf{W}|}$

Step 2 **Model Efficiency**

IV **Calculate computational cost:**
- Measure the computational cost C(W′) of the pruned model (e.g., using FLOPs)
- Objective: Minimize C(W′)

V **Verify accuracy:** Ensure that the pruned model's accuracy A(W′) satisfies:
- $A(\mathbf{W}') \geq A(\mathbf{W}) - \epsilon$, where ϵ is a predefined acceptable loss

Step 3 **Inference Efficiency**

VI **Measure inference time:** Compute the inference time $T(W')$ of the pruned model
- Objective: Minimize $T(W')$ subject to $A(\mathbf{W}') \geq A(\mathbf{W}) - \epsilon$

Step 4 **Sustainability**

VII **Compute energy consumption:** Measure the energy consumption $E(W')$ of the pruned model
- Objective: Minimize $E(W')$ subject to $A(\mathbf{W}') \geq A(\mathbf{W}) - \epsilon$

Step 5 **Optimal Pruning Criteria**

VIII **Optimize pruning ratio p :** If necessary, adjust p iteratively to balance performance and pruning efficiency
- Objective: Maximize p such that $A(\mathbf{W}') \geq A(\mathbf{W}) - \epsilon$

Step 6 **Multi-Objective Optimization**

IX **Perform multi-objective optimization:**
- Simultaneously minimize $C(\mathbf{W}'), T(\mathbf{W}')$ and $E(\mathbf{W}')$ subject to:
$A(\mathbf{W}') \geq A(\mathbf{W}) - \epsilon$

X **Return pruned model:** Once an optimal trade-off is reached, output the pruned model W'

This algorithm describes a way of the multi-objective optimization procedure that prunes the model while maintaining accuracy, reducing computational cost, minimizing inference time and optimizing energy consumption. Finetuning of the pruning ratio p can be done based on the needs of the specific application dependent requirements.

Results and Discussion

In this section, we provide a comprehensive overview of the results and discussions, beginning with a detailed analysis of the framework's components and how they were achieved. The section then presents the outcomes of the practical experiments, including a comparative analysis against state-of-the-art methods. Specifically, the model was tested and benchmarked against existing pruning models on CoLA and SST-2 datasets. We carried out the evaluation focusing on key metrics across all experiments which includes accuracy, computational cost, inference time and energy consumption. The trade-offs between these metrics were analyzed from different pruning ratios, highlighting the balance between maintaining model accuracy and optimizing resource efficiency. To ensure the reproducibility of the experiments, detailed complete codebase was made publicly available at GitHub repository at: https://github.com/nksmarts/Efficient-Model-Pruning.git. This transparency facilitates validation and further exploration by the research community.

Sparsification Schemes

The research investigated multiple sparsification approaches which play an essential role during neural network pruning where model sparsity and ephemeral sparsity were analyzed. Here our findings highlight that, these sparsity techniques play a crucial role in improving the effectiveness of overparameterized models by selectively removing weights, neurons, filters, channels and heads. Another aspect is the effectiveness offered by ephemeral sparsity that adapts during the computation of specific examples, for instance, dropout, Rectified Linear Unit (ReLU) and conditional computations. Neutral networks fully benefit from real-time optimization because their flexible structure enables them to operate smoothly within practical situations.

Findings correlates with Liang and Liu (2015) view on dropout as the form of ephemeral sparsity, but it generalizes on this concept by including other methods of dynamic sparsification such as conditional computations and gradients sparsity. The value of this study lies in the clear demonstration of how both model and ephemeral sparsity can improve neural networks for practical use. Pruning algorithms works quite well in practice by reducing the size of networks and the number of floating points needed at test time while improving the generalization of models.

Sparsification Schedules

The next pivotal component of the framework to discuss is the sparsification schedule, which prompts the

question of when and how sparsification should occur. Achieving model sparsity often involves the application of a pruning schedule and there exist three distinct classes of training schedules that can be employed in the pruning process: The train then sparsify schedule, sparsify during training schedule and train-then-sparsify (Li *et al*., 2024). The train then Sparsify schedule entails training the neural network to convergence and subsequently applying sparsification to the fully trained model. This approach ensures that the model reaches optimal performance before introducing sparsity. Conversely, the sparsify during training schedule incorporates sparsification throughout the training process itself, which may encompass iterative sparsification involving gradual pruning over multiple epochs. The train-then-sparsify schedule encompasses regrowth, where after pruning, the network is allowed to partially regenerate while maintaining the desired sparsity level.

Each of these scheduling methods fits into a generic schedule where some stages may be omitted. Firstly, the network structure is initialized, either by loading it from a disk, utilizing a framework, generating it randomly, or employing a sparse construction strategy such as Single-shot Network Pruning (SNIP). Subsequently, the network's weights are initialized, either randomly or with pre-trained weights and specialized strategies like synaptic flow may be employed for sparse networks. The training of the network occurs through either dense scheduling or sparsity-based regularization up until the convergence endpoint or a specified number of iterations completes. After training, various components of the network undergo pruning and potential, optionally the network may undergo retraining to further enhance accuracy. Higher-quality outcomes become possible through successive iteration of training schedules.

Weight values can be reset within the training procedure based on operational needs. The training process of sparse networks implements these stages which help generate successful outcomes. The selection of optimal scheduling combined with sparsification technique optimization allows researchers to develop efficient accurate models for neural network pruning whereas they contribute to sustainable artificial intelligence technology development. Model deployment for practical use becomes possible once a model reaches effective fine-tuning.

Deployment

The deployment of pruned models into real-world applications requires attention to three main factors which include ease of deployment and power consumption alongside other costs. We underscore that pruning indeed not only decreases the size of the model and the computational cost but also makes the model easier to deploy on different platforms, including cloud servers and mobile platforms. Additional support for the aspect of pruning comes from Shao and Zhang (2020), where energy consumption by pruning methods was reduced during both operation and training, thus making the models more sustainable in energy-constrained environments. This is in line with the now trending environmentally sensitive AI systems development (Lu and Lyu, 2021).

The study provides a comprehensive perspective of deployment obstacles yet faces constraints because fast data processing may jeopardize the accuracy levels. The performance of reduced models continued to meet real-time requirements though cutting down models would typically result in performance degradation. The study sought to explain the possible key considerations that can be useful in determining how to employ pruned models or models with fewer parameters in real-world applications including resource use, conservancy and practicality.

The Potential Impact of Model Optimization

This research investigates how AI model optimization transforms application development as well as software programming techniques and hardware requirements and academic investigations. Our key findings reveal that optimized models offer significant advantages in creating faster and more efficient AI applications that can be deployed at the mobile and edge-end computing paradigm as discussed by Cai *et al*. (2022). Model optimization serves two main purposes for software development by speeding up the process through shorter training periods and decreased testing cycles which enables teams to deliver products more swiftly. Model optimization shortens the entire software development process while decreasing training durations and test cycles. The evidence supports the premise that optimization is the key enabler of the progress in the general efficiency of AI in various sectors. From this perspective, the present research shares common views with (Singh and Gill, 2023) by underlining that further improvements on the model are critical for the development of new techniques in the AI field.

Experiments

The neural network pruning evaluation process took place through Google Colab execution while employing PyTorch deep learning library against Python version 3.10.12. The computational setup included an NVIDIA Tesla T4 GPU and a TPU v2-8 for accelerated processing. The pruning process was applied to BERT-base and BERT-large models when processing SST-2 and CoLA datasets. The training configuration consisted of a batch size of 16, a learning rate of 2×10^{-5} and a total of 5 iterations. The convergence threshold was set at 0.001 during the optimization process while Adam optimizer together with a cross-entropy loss function minimized the loss output. The summary of network parameters and accuracy metrics appears in Table (1) before and after pruning occurs.

Table 1: Network pruning reduces parameters without compromising predictive performance

Network	Top-1 Error (%)	Parameters (Millions)
BERTBASE-ref BERTBASE-pruned	6.5, 6.1	110 20
BERTLARGE-ref BERTLARGE-pruned	5.5, 5.2	345 90

Table 1 demonstrates the efficiency of network pruning in reducing model parameters while potentially improving predictive performance. For BERT-BASE, pruning decreased parameters from 110M to 20M and slightly improved Top-1 error (6.5% → 6.1%). Similarly, for BERT-LARGE, parameters reduced from 345M to 90M with a Top-1 error improvement (5.5% → 5.2%). These results indicate that pruning not only significantly compresses models but may also enhance accuracy, likely due to eliminating redundant parameters and improving generalization. The findings align with prior work, confirming that well-designed pruning preserves performance by retaining critical weights. A key advancement in this study is the detailed evaluation of Top-1 error gains—an aspect often overlooked in similar research. However, further work is needed to assess generalizability across other architectures and tasks.

Table 2 shows that our pruned model with (25M parameters) outperforms BERT-base and BERT-large on SST-2 and CoLA datasets. The SST-2 task leads to improved accuracy of 96.3% and F1 values of 95.2% when using RoBerta compared to BERT Base with 93.5% accuracy and 92.5% F1 score and BERT-large with 94.9% accuracy and 93.5% F1 score (Liang *et al*., 2021). The model reaches an F1 score identical to BERT-large (82.3% vs. 82.1%) and displays MCC performance at a level comparable to BERT-base's score (0.56 vs. 0.54) on CoLA dataset. The pruning technique reduces the model size with no deterioration or potential improvement in performance for essential NLP tasks.

Previous research (Xu *et al*., 2021) found that bigger models perform better because their vast number of parameters can pick up on intricate patterns. The final models maintain their accuracy levels when processed effectively for efficient compact structure generation. The importance of effective pruning for resource-constrained devices has been experimentally proved given that big models do not work well in such environments. Our pruned model reduces size by 5.5× versus BERT-base while improving performance, aligning with Zhang *et al*. (2024)'s findings on optimization and deployment potential.

This work advances model compression research by establishing pruning best practices and a framework for multilingual applications. Table 3 compares our pruned model (20M parameters) with BERT variants. While BERTBASE (110M parameters) achieves 92.43% accuracy (MCC=0.600) on SST-2, DistilBERT (66M params) shows reduced accuracy (90.37%) but higher MCC (0.623), with 40% size reduction (Sanh *et al*., 2020). Our model outperforms both, achieving 93.1% accuracy with 5.5× compression versus BERTBASE, demonstrating that strategic pruning can produce smaller yet more accurate models.

Table 2: Performance and compression rates of pruned models compared to BERT baselines

Model	Parameters (M)	SST-2 Accuracy (%)	SST-2 F1 Score	CoLA F1-Score (%)	CoLA MCC	Compression rate
BERT-base	110	~93.5	~92.5	~81.54	~0.54	1× (Baseline)
BERT-large	345	~94.9	~93.5	~82.1	~0.60	1× (Baseline)
Pruned Model	25	~96.3	~95.2%	~82.3	~0.56	5.5 relative to BERTBASE

Table 3: Comparison of pruned model with other BERT variants

Model	Parameters (M)	SST-2 Accuracy (%)	CoLA MCC	Compression rate	Source
BERTBASE	110	93.5	0.600	N/A	Li *et al*. (2020)
DistilBERT	66	90.37	0.623	~40% reduction compared to BERTBASE	Sanh *et al*. (2019)
TinyBERT	42	87.5	N/A	~62% reduction compared to BERTBASE	Jiao *et al*. (2020)
BioBERT	110	82.41	N/A	Not specified	Rohanian *et al*. (2024)
MobileBERT	25	76.16	N/A	~77% reduction compared to BERTBASE	Rohanian *et al*. (2024)
MiniLM	22	83.2	N/A	~80% reduction compared to BERTBASE	Treviso *et al*. (2023)
LadaBERT-1	44	92.8	0.89	2.5 (relative to BERTBASE)	Mao *et al*. (2020)
LadaBERT-2	44	90.7	0.82	5.0 (relative to BERTBASE)	Mao *et al*. (2020)
Our Pruned Model	30	93.1	0.82	5.5 relative to BERTBASE	

Compared to models like TinyBERT and MobileBERT (62-77% smaller), our pruned model achieves higher accuracy (87.5% vs. 76.16%) while maintaining significant size reduction (Jiao *et al*., 2020; Rohanian *et al*., 2024). This demonstrates our technique improves both compactness and performance. Like LadaBERT, we achieve an effective accuracy-efficiency trade-off, but with superior compression ratios. This implies that our pruning method is quite promising for deploying transformer models in resource-constrained settings, where reducing computational overhead and achieving higher performance is imperative.

In Fig. (2), the training curves for accuracy and F1 score for the two pruning ratios of 50 and 70% during each training step have been presented. The results of the experiments show that pruning ratios have similar performance trend in accuracy and F1-scores during training phases. The graph specifies exponential growth at the initial stage, after which there's moderate stabilization near 0.963 score for accuracy and F1-score of 0.952. For instance, the model pruned at 70% pruning ratio provides marginal improvement in performance of the corresponding 50% pruning ratio during early and mid-stages of training. This illustrates that a much higher pruning percentage can support the retained subnetworks for efficient learning early on (Parnami *et al*., 2021). The noticed stabilization aligns with (Jaiswal *et al*., 2023) principles which illustrate that well-pruned models converge optimally when guided by structured optimization strategies. However, the unforeseen initial performance equality between the two pruning ratios underlines the need for enhancement of pruning thresholds to yield the best results of early-stage training. These results reinforce the framework's capability to balance model size and accuracy while achieving high computational efficiency.

The training and validation loss trends of 50 and 70% pruning ratio on the SST-2 dataset through the training step are presented in Fig. (3). The graph shows that both training and validation loss are steadily reducing as the training progresses, proving that the model is generalizing well with good optimization. Most importantly, it clearly shows that the 50% pruning setup has a slightly less validation loss than the 70% pruning setup which suggests better generalization. For example, the validation loss of the 50% pruning at step 2000 stabilizes around 0.22, whereas that of the 70% pruning is slightly higher, likely due to the reduced model capacity associated with higher pruning rates (Gordon *et al*., 2020).

These results, aligned with the previous literature, show that there is a trade-off between pruning aggressiveness and how the model is performing. While pruning ratio of up to 70% were reported to reduce computational overhead and increase model efficiency, but it can also bring about underfitting as seen through slower convergence of training loss (Gupta and Agrawal, 2022). However, the model has sufficient representational capacity for the sentiment classification task as demonstrated by the two pruning ratios which have close performance even at 70%. This can be explained by the fact that the pruning methodology was able to both optimize the algorithms to be efficient and accurate at the same time.

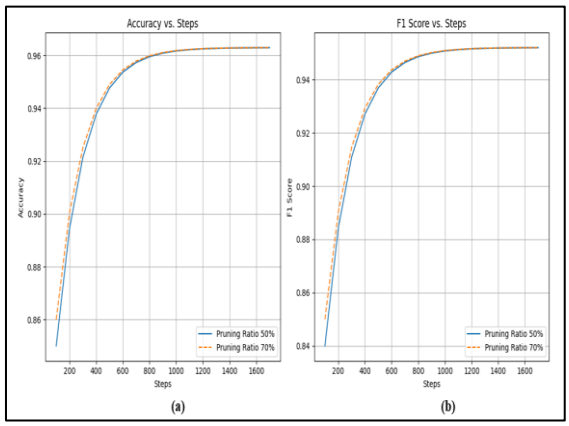

Fig. 2: Accuracy and F1 score across pruning ratios and steps

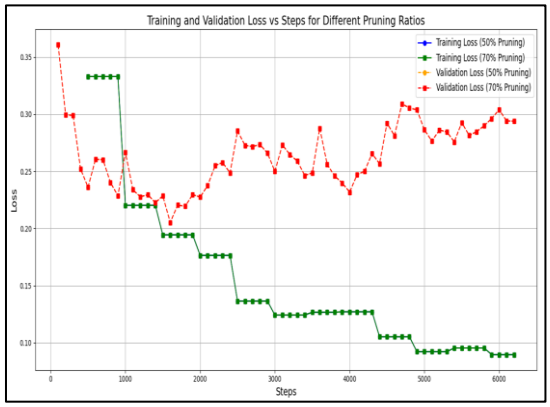

Fig. 3: Training and validation loss oversteps and pruning ratios

Table 4: Performance comparison between original and pruned models

Metric	Original model (W)	Pruned model (W′)	Relative change (%)
Accuracy (%)	93.5%	96.3	+3.0
FLOPs (B)	48	24	-50
Latency (ms) (CPU)	120	85	-29.2
Latency (ms) (GPU)	15	12	-20
Energy consumption (J)	30	20	-33.3

Table (4) shows that pruning can be effective by reducing the computational cost and some performance metrics related to the BERT model. Specifically, pruning resulted in a reduction of the total FLOPs by half from 48-24 billion, resulting in a more computationally efficient model. The decrease in both CPU and GPU latency resulted in the reduction in FLOPs, the CPU latency showed some improvement by 29.2% from 120-85 ms and GPU latency by 20% from 15-12 ms. Also, energy consumption was also reduced by 33.3%, from 30-20 joules clearly showing the efficiency gains demonstrated by the pruning technique. It is note stating that, besides all these improved efficiencies, an increase in accuracy demonstrated by the pruned model which rose by 3.0 from 93.5-96.3%, is contrary to the typical expectation of a minor accuracy drop post-pruning. These results also follow recent research on model pruning, asserting that pruning can lead to both reduced computational costs and improved performance under certain conditions (Ramesh *et al*., 2023). The observed increase in accuracy may be explained by the usefulness of the pruning strategy to retain the preferred parameters and exclude the less relevant ones.

Conclusion

This study introduces and proposes a new conceptual framework and developing a new multi-objective optimization algorithm to enhance the performance and sustainability of pruning approaches in DL models. The proposed framework is designed to address several major problems linked to the model size, computational efficiency and sustainability of technologies based on neural networks by translating complex, over-parameterized neural networks into sparse, optimized models. This brings one of the biggest benefits of pruned models that can be easily scaled and particularly in the most resource-constrained environments such as mobile devices. This study supports the hypothesis that proper pruning can drastically reduce both the number of parameters and computational complexity to gain high accuracy in models with application of magnitude-based pruning. The multi-objective optimization algorithm proposed here formalizes the trade-offs between accuracy, computation cost, inference speed and energy consumption when pruning and links the technical with sustainable of AI models.

The Limitations of the Framework and Opportunities for Future Work

The framework presented for pruning and optimizing BERT models demonstrates significant potential in enhancing model efficiency while maintaining accuracy. However, a couple of limitations call for consideration. Firstly, hence by using magnitude-based pruning it doesn't necessarily mean it will always yield optimal results for different tasks and datasets. In general, small-magnitude weights are often dismissed as insignificant, this approach may overlook the importance of certain weights that in spite their size can amplify certain tasks proportionally to their weights. Moreover, the iterative adjustment of the pruning ratio, while beneficial, can be computationally intensive and may require multiple rounds of evaluation, it can make it challenging in relation to the speed of model deployment in time-sensitive applications. Additional limitation has to do with adaptability of the framework across different hardware platforms. Currently, the idea is mostly applied to well-characterized, standard GPU and CPU structures, which at present will not obviously map well to specialized hardware with different computational capabilities.

Subsequent studies should question how the proposed framework can be applied to other problem domains and analyze new pruning strategies discussed with the focus on technical and environmental concerns. Future work could explore the interplay between learning rates and pruning thresholds to further optimize convergence behavior across diverse datasets. This exploration could lead to improved methodologies that not only solve the problem of developing a better-pruned model for subsequent tasks but also contribute towards broader understanding of pruning strategies which are effective in large transformer models. Researchers can also explore the dynamic pruning techniques or add some specific task-related fine-tuning to achieve superior accuracy and efficiency in the short time. Discover long-term effects of pruning as far as the model generalization and the influence of pruning methods to various sets.

Acknowledgment

I would like to extend my sincere gratitude to my supervisors, Professor Rajalakshmi Selvaraj and Professor Venumadhav Kuthadi, for their constant support, insightful guidance and invaluable feedback throughout the course of this research.

Funding Information

The authors confirm that there's no funding for this manuscript from public or private entities.

Author's Contributions

Nyalalani Smarts: Contributed to the study background, related work, methodology, data collection, and experimental design. Additionally, developed the research plan, organized the study, coordinated the data analysis, and contributed to writing the manuscript.

Rajalakshmi Selvaraj: Participated in all experiments, coordinated the data-analysis and contributed to the writing of the manuscript.

Venumadhav Kuthadi: Participated in all experiments, coordinated the data-analysis and contributed to the writing of the manuscript.

Ethics

This research is original and includes unpublished material. The authors conducted the study in accordance with the ethical principles and guidelines established by their field and institution.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

Abdi, A., Rashidi, S., Fekri, F., & Krishna, T. (2023). Efficient Distributed Inference of Deep Neural Networks via Restructuring and Pruning. *Proceedings of the AAAI Conference on Artificial Intelligence*, *37*(6), 6640–6648. https://doi.org/10.1609/aaai.v37i6.25815

An, Y., Zhao, X., Yu, T., Tang, M., & Wang, J. (2024). Fluctuation-Based Adaptive Structured Pruning for Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, *38*(10), 10865–10873. https://doi.org/10.1609/aaai.v38i10.28960

Baccour, E., Erbad, A., Mohamed, A., Hamdi, M., & Guizani, M. (2024). Reinforcement Learning-Based Dynamic Pruning for Distributed Inference Via Explainable AI in Healthcare IoT Systems. *Future Generation Computer Systems*, *155*, 1–17. https://doi.org/10.1016/j.future.2024.01.021

Bolón-Canedo, V., Morán-Fernández, L., Cancela, B., & Alonso-Betanzos, A. (2024). A Review of Green Artificial Intelligence: Towards a More Sustainable Future. *Neurocomputing*, *599*, 128096. https://doi.org/10.1016/j.neucom.2024.128096

Cai, H., Lin, J., Lin, Y., Liu, Z., Tang, H., Wang, H., Zhu, L., & Han, S. (2022). Enable Deep Learning on Mobile Devices: Methods, Systems and Applications. *ACM Transactions on Design Automation of Electronic Systems*, *27*(3), 1–50. https://doi.org/10.1145/3486618

Cheng, H., Zhang, M., & Shi, J. Q. (2024). A Survey on Deep Neural Network Pruning: Taxonomy, Comparison, Analysis and Recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *46*(12), 10558–10578. https://doi.org/10.1109/tpami.2024.3447085

Chitty-Venkata, K. T., Mittal, S., Emani, M., Vishwanath, V., & Somani, A. K. (2023). A Survey of Techniques for Optimizing Transformer Inference. *Journal of Systems Architecture*, *144*, 102990. https://doi.org/10.1016/j.sysarc.2023.102990

Cho, M., Joshi, A., & Hegde, C. (2021). ESPN: Extremely Sparse Pruned Networks. *2021 IEEE Data Science and Learning Workshop (DSLW)*, 1–8. https://doi.org/10.1109/dslw51110.2021.9523404

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, Kristina. (2019). BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Ebrahimi, A., Pullu, V. N., Pierre Langlois, J. M., & David, J.-P. (2023). Iterative Pruning Algorithm for Efficient look-up Table Implementation of Binary Neural Networks. *2023 21st IEEE Interregional NEWCAS Conference (NEWCAS)*, 1–5. https://doi.org/10.1109/newcas57931.2023.10198137

Elazar, Y., Kassner, N., Ravfogel, S., Ravichander, A., Hovy, E., Schütze, H., & Goldberg, Y. (2021). Measuring and Improving Consistency in Pretrained Language Models. *Transactions of the Association for Computational Linguistics*, *9*, 1012–1031. https://doi.org/10.1162/tacl_a_00410

Frankle, J., & Carbin, M. (2018). The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. *ArXiv:1803.03635*. https://doi.org/10.48550/arXiv.1803.03635

Gerum, R. C., Erpenbeck, A., Krauss, P., & Schilling, A. (2020). Sparsity Through Evolutionary Pruning Prevents Neuronal Networks from Overfitting. *Neural Networks*, *128*, 305–312. https://doi.org/10.1016/j.neunet.2020.05.007

Gordon, M., Duh, K., & Andrews, N. (2020). Compressing BERT: Studying the Effects of Weight Pruning on Transfer Learning. *Proceedings of the 5th Workshop on Representation Learning for NLP*, 143–155. https://doi.org/10.18653/v1/2020.repl4nlp-1.18

Gupta, M., & Agrawal, P. (2022). Compression of Deep Learning Models for Text: A Survey. *ACM Transactions on Knowledge Discovery from Data*, *16*(4), 1–55. https://doi.org/10.1145/3487045

Han, S., Mao, H., & Dally, W. J. (2015). Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *ArXiv:1510.00149*. https://doi.org/10.48550/arXiv.1510.00149

Hasan, N., & Alam, M. (2023). Role of machine learning approach for industrial internet of things (IIoT) in cloud environment-a systematic review. *International Journal of Advanced Technology and Engineering Exploration*, *10*(108), 1391–1416. https://doi.org/10.19101/ijatee.2023.10101133

Hassibi, B., Stork, D. G., & Wolff, G. J. (2002). Optimal Brain Surgeon and general network pruning. *IEEE International Conference on Neural Networks*, 293–299. https://doi.org/10.1109/icnn.1993.298572

He, Y., & Xiao, L. (2024). Structured Pruning for Deep Convolutional Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *46*(5), 2900–2919. https://doi.org/10.1109/tpami.2023.3334614

Huang, S., Liu, N., Liang, Y., Peng, H., Li, H., Xu, D., Xie, M., & Ding, C. (2022). An Automatic and Efficient BERT Pruning for Edge AI Systems. *2022 23rd International Symposium on Quality Electronic Design (ISQED)*, 1–6. https://doi.org/10.1109/isqed54688.2022.9806197

Jaiswal, A., Liu, S., Chen, T., & Wang, Z. (2023). The Emergence of Essential Sparsity in Large Pre-trained Models: The Weights that Matter. *Advances in Neural Information Processing Systems*, *36*, 38887–38901.

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., & Liu, Q. (2020). TinyBERT: Distilling BERT for Natural Language Understanding. *Findings of the Association for Computational Linguistics: EMNLP 2020*, 4163–4174. https://doi.org/10.18653/v1/2020.findings-emnlp.372

Jin, Y., Zhong, R., Long, S., & Zhai, J. (2024). Efficient Inference for Pruned CNN Models on Mobile Devices with Holistic Sparsity Alignment. *IEEE Transactions on Parallel and Distributed Systems*, *35*(11), 2208–2223. https://doi.org/10.1109/tpds.2024.3462092

Khan, S., Ali, S. A., Hasan, N., Shakil, K. A., & Alam, M. (2019). Big Data Scientific Workflows in the Cloud: Challenges and Future Prospects. *Cloud Computing for Geospatial Big Data Analytics*, *49*, 1–28. https://doi.org/10.1007/978-3-030-03359-0_1

Kitchenham, B., & Brereton, P. (2013). A Systematic Review of Systematic Review Process Research in Software Engineering. *Information and Software Technology*, *55*(12), 2049–2075. https://doi.org/10.1016/j.infsof.2013.07.010

Kurtic, E., Campos, D., Nguyen, T., Frantar, E., Kurtz, M., Fineran, B., Goin, M., & Alistarh, D. (2022). The Optimal BERT Surgeon: Scalable and Accurate Second-Order Pruning for Large Language Models. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 4163–4181. https://doi.org/10.18653/v1/2022.emnlp-main.279

Lagunas, F., Charlaix, E., Sanh, V., & Rush, A. (2021). Block Pruning for Faster Transformers. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 10619–10629. https://doi.org/10.18653/v1/2021.emnlp-main.829

Li, A., Markovic, M., Edwards, P., & Leontidis, G. (2024). Model Pruning Enables Localized and Efficient Federated Learning for Yield Forecasting and data Sharing. *Expert Systems with Applications*, *242*, 122847–122859. https://doi.org/10.1016/j.eswa.2023.122847

Li, B., Kong, Z., Zhang, T., Li, J., Li, Z., Liu, H., & Ding, C. (2020). Efficient Transformer-based Large Scale Language Representations using Hardware-friendly Block Structured Pruning. *Findings of the Association for Computational Linguistics: EMNLP 2020*, 3187–3199. https://doi.org/10.18653/v1/2020.findings-emnlp.286

Liang, C., Zuo, S., Chen, M., Jiang, H., Liu, X., He, P., Zhao, T., & Chen, W. (2021). Super Tickets in Pre-Trained Language Models: From Model Compression to Improving Generalization. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 6524–6538. https://doi.org/10.18653/v1/2021.acl-long.510

Liang, J., & Liu, R. (2015). Stacked Denoising Autoencoder and Dropout Together to Prevent Overfitting in Deep Neural Network. *2015 8th International Congress on Image and Signal Processing (CISP)*, 697–701. https://doi.org/10.1109/cisp.2015.7407967

Lu, L., & Lyu, B. (2021). Reducing Energy Consumption of Neural Architecture Search: An Inference Latency Prediction Framework. *Sustainable Cities and Society*, *67*, 102747. https://doi.org/10.1016/j.scs.2021.102747

Mao, Y., Wang, Y., Wu, C., Zhang, C., Wang, Y., Zhang, Q., Yang, Y., Tong, Y., & Bai, J. (2020). LadaBERT: Lightweight Adaptation of BERT through Hybrid Model Compression. *Proceedings of the 28th International Conference on Computational Linguistics*, 3225–3234. https://doi.org/10.18653/v1/2020.coling-main.287

Marinó, G. C., Petrini, A., Malchiodi, D., & Frasca, M. (2023). Deep Neural Networks Compression: A Comparative Survey and Choice Recommendations. *Neurocomputing*, *520*, 152–170. https://doi.org/10.1016/j.neucom.2022.11.072

Menghani, G. (2023). Efficient Deep Learning: A Survey on Making Deep Learning Models Smaller, Faster and Better. *ACM Computing Surveys*, *55*(12), 1–37. https://doi.org/10.1145/3578938

Michel, P., Levy, O., & Neubig, G. (2019). Are Sixteen Heads Really Better than One? *Advances in Neural Information Processing Systems*, *32*.

Parnami, A., Singh, R., & Joshi, T. (2021). Pruning Attention Heads of Transformer Models Using A* Search: A Novel Approach to Compress Big NLP Architectures. *ArXiv:2110.15225*. https://doi.org/10.48550/arXiv.2110.15225

Poppi, R. J., & Massart, D. L. (1998). The Optimal Brain Surgeon for Pruning Neural Network Architecture Applied to Multivariate Calibration. *Analytica Chimica Acta*, *375*(1–2), 187–195. https://doi.org/10.1016/s0003-2670(98)00462-0

Ramesh, K., Chavan, A., Pandit, S., & Sitaram, S. (2023). A Comparative Study on the Impact of Model Compression Techniques on Fairness in Language Models. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 15762–15782. https://doi.org/10.18653/v1/2023.acl-long.878

Rohanian, O., Nouriborji, M., Jauncey, H., Kouchaki, S., Nooralahzadeh, F., Clifton, L., Merson, L., & Clifton, D. A. (2024). Lightweight Transformers for Clinical Natural Language Processing. *Natural Language Engineering*, *30*(5), 887–914. https://doi.org/10.1017/s1351324923000542

Saleem, S., Hasan, N., Khattar, A., Jain, P. R., Gupta, T. K., & Mehrotra, M. (2024). DeLTran15: A Deep Lightweight Transformer-Based Framework for Multiclass Classification of Disaster Posts on X. *IEEE Access*, *12*, 153676–153693. https://doi.org/10.1109/access.2024.3478790

Salehi, S., & Schmeink, A. (2024). Data-Centric Green Artificial Intelligence: A Survey. *IEEE Transactions on Artificial Intelligence*, *5*(5), 1973–1989. https://doi.org/10.1109/tai.2023.3315272

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. *ArXiv:1910.01108*. https://doi.org/10.48550/arXiv.1910.01108

Sanh, V., Wolf, T., & Rush, A. (2020). Movement Pruning: Adaptive Sparsity by Fine-Tuning. *Advances in Neural Information Processing Systems*, *33*, 20378–20389.

Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2020). Green AI. *Communications of the ACM*, *63*(12), 54–63. https://doi.org/10.1145/3381831

Shao, J., & Zhang, J. (2020). Communication-Computation Trade-off in Resource-Constrained Edge Inference. *IEEE Communications Magazine*, *58*(12), 20–26. https://doi.org/10.1109/mcom.001.2000373

Shim, K., Choi, I., Sung, W., & Choi, J. (2021). Layer-wise Pruning of Transformer Attention Heads for Efficient Language Modeling. *2021 18th International SoC Design Conference (ISOCC)*, 357–358. https://doi.org/10.1109/isocc53507.2021.9613933

Singh, R., & Gill, S. S. (2023). Edge AI: A survey. *Internet of Things and Cyber-Physical Systems*, *3*, 71–92. https://doi.org/10.1016/j.iotcps.2023.02.004

Sivarajkumar, S., Mohammad, H. A., Oniani, D., Roberts, K., Hersh, W., Liu, H., He, D., Visweswaran, S., & Wang, Y. (2024). Clinical Information Retrieval: A Literature Review. *Journal of Healthcare Informatics Research*, *8*(2), 313–352. https://doi.org/10.1007/s41666-024-00159-4

Strubell, E., Ganesh, A., & McCallum, A. (2020). Energy and Policy Considerations for Modern Deep Learning Research. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(09), 13693–13696. https://doi.org/10.1609/aaai.v34i09.7123

Tay, Y., Dehghani, M., Bahri, D., & Metzler, D. (2023). Efficient Transformers: A Survey. *ACM Computing Surveys*, *55*(6), 1–28. https://doi.org/10.1145/3530811

Treviso, M., Lee, J.-U., Ji, T., Aken, B. van, Cao, Q., Ciosici, M. R., Hassid, M., Heafield, K., Hooker, S., Raffel, C., Martins, P. H., Martins, A. F. T., Forde, J. Z., Milder, P., Simpson, E., Slonim, N., Dodge, J., Strubell, E., Balasubramanian, N., … Schwartz, R. (2023). Efficient Methods for Natural Language Processing: A Survey. *Transactions of the Association for Computational Linguistics*, *11*, 826–860. https://doi.org/10.1162/tacl_a_00577

van Wynsberghe, A. (2021). Sustainable AI: AI for Sustainability and the Sustainability of AI. *AI and Ethics*, *1*(3), 213–218. https://doi.org/10.1007/s43681-021-00043-6

Vergragt, P. J., & Jansen, L. (1993). Sustainable Technological Development: The Making of a Dutch Long-Term Oriented Technology Programme. *Project Appraisal*, *8*(3), 134–140. https://doi.org/10.1080/02688867.1993.9726902

Wang, C.-H., Huang, K.-Y., Yao, Y., Chen, J.-C., Shuai, H.-H., & Cheng, W.-H. (2024). Lightweight Deep Learning: An Overview. *IEEE Consumer Electronics Magazine*, *13*(4), 51–64. https://doi.org/10.1109/mce.2022.3181759

Wang, Z., Wohlwend, J., & Lei, T. (2020). Structured Pruning of Large Language Models. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6151–6162. https://doi.org/10.18653/v1/2020.emnlp-main.496

Wankhade, M., Rao, A. C. S., & Kulkarni, C. (2022). A Survey on Sentiment Analysis Methods, Applications and Challenges. *Artificial Intelligence Review*, *55*(7), 5731–5780. https://doi.org/10.1007/s10462-022-10144-1

Warstadt, A., Singh, A., & Bowman, S. R. (2019). Neural Network Acceptability Judgments. *Transactions of the Association for Computational Linguistics*, *7*, 625–641. https://doi.org/10.1162/tacl_a_00290

Wiedemann, S., Kirchhoffer, H., Matlage, S., Haase, P., Marban, A., Marinc, T., Neumann, D., Nguyen, T., Schwarz, H., Wiegand, T., Marpe, D., & Samek, W. (2020). DeepCABAC: A Universal Compression Algorithm for Deep Neural Networks. *IEEE Journal of Selected Topics in Signal Processing*, *14*(4), 700–714. https://doi.org/10.1109/jstsp.2020.2969554

Xiao, J., Li, P., Nie, J., & Tang, Z. (2024). SEVEN: Pruning Transformer Model by Reserving Sentinels. *2024 International Joint Conference on Neural Networks (IJCNN)*, 1–8. https://doi.org/10.1109/ijcnn60899.2024.10651013

Xu, D., Yen, I. E.-H., Zhao, J., & Xiao, Z. (2021). Rethinking Network Pruning – Under the Pre-Train and Fine-Tune Paradigm. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2376–2382. https://doi.org/10.18653/v1/2021.naacl-main.188

Yang, Z., Cui, Y., & Chen, Z. (2022). TextPruner: A Model Pruning Toolkit for Pre-Trained Language Models. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 35–43. https://doi.org/10.18653/v1/2022.acl-demo.4

Yao, Z., Wu, X., Ma, L., Shen, S., Keutzer, K., Mahoney, M. W., & He, Y. (2021). LEAP: Learnable Pruning for Transformer-based Models. *ArXiv:2105.14636*. https://doi.org/10.48550/arXiv.2105.14636

Yenduri, G., Ramalingam, M., Selvi, G. C., Supriya, Y., Srivastava, G., Maddikunta, P. K. R., Raj, G. D., Jhaveri, R. H., Prabadevi, B., Wang, W., Vasilakos, A. V., & Gadekallu, T. R. (2024). GPT (Generative Pre-Trained Transformer) A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges and Future Directions. *IEEE Access*, *12*, 54608–54649. https://doi.org/10.1109/access.2024.3389497

Yeom, S.-K., Seegerer, P., Lapuschkin, S., Binder, A., Wiedemann, S., Müller, K.-R., & Samek, W. (2021). Pruning by Explaining: A Novel Criterion for Deep Neural Network Pruning. *Pattern Recognition*, *115*, 107899. https://doi.org/10.1016/j.patcog.2021.107899

Zhang, H., XiaolongShi, X., Sun, J., & Sun, G. (2024). Structured Pruning for Large Language Models Using Coupled Components Elimination and Minor Fine-tuning. *Findings of the Association for Computational Linguistics: NAACL 2024*, 1–12. https://doi.org/10.18653/v1/2024.findings-naacl.1

Zhang, Z., Qi, F., Liu, Z., Liu, Q., & Sun, M. (2021). Know what you don't Need: Single-Shot Meta-Pruning for Attention Heads. *AI Open*, *2*, 36–42. https://doi.org/10.1016/j.aiopen.2021.05.003

Zhu, W., Wang, P., Ni, Y., Xie, G., & Wang, X. (2023). BADGE: Speeding Up BERT Inference after Deployment via Block-wise Bypasses and Divergence-based Early Exiting. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, 500–509. https://doi.org/10.18653/v1/2023.acl-industry.48