Optimizing Transformer Models for Energy Efficiency in Time Series Classification

Green Computing Research Project

Instructor: Dr. Ziliang Zong

TYPE THE PAPER AUTHOR NAME HERE

Written by: Habib Irani

Optimizing Transformer Models for Energy Efficiency in Time Series Classification

Green Computing Research Project

Abstract

In the rapidly evolving landscape of artificial intelligence (AI), the development of energy-efficient and environmentally sustainable models has emerged as a critical challenge, especially for transformer models utilized in time series classification. This project aims to address the substantial energy consumption and carbon footprint associated with training and deploying these models. By focusing on optimization techniques such as pruning, quantization, and knowledge distillation, specifically tailored to the domain of time series data, the research endeavors to enhance the computational efficiency and environmental sustainability of transformer models. Employing the UniMiB SHAR dataset for human activity recognition as a testbed, this study will systematically evaluate the impact of various optimization strategies on model performance, energy consumption, and speed. Based on my observations by applying optimization techniques, we can get better results in terms of time and energy consumption with not so much decreasing in accuracy[1].

1. Introduction

The rapid ascent of artificial intelligence (AI) technologies, particularly in the field of time series classification, has underscored the transformative potential of these tools across a myriad of domains, from healthcare diagnostics to

[1] Project source code: https://github.com/habibirani/Green-Computing-Project

financial forecasting. At the heart of these advancements lie transformer models, lauded for their superior ability to discern and leverage patterns within sequential data. Yet, as these models increasingly become the cornerstone of AI-driven solutions, their substantial energy demands and resultant carbon emissions have sparked significant concern. This burgeoning environmental toll, particularly pronounced in real-time analysis and prediction tasks characteristic of time series classification, presents a critical challenge that must be addressed.

This research project is poised at the intersection of technological innovation and environmental stewardship, aiming to reconcile the two by optimizing transformer models for enhanced energy efficiency in time series classification. Drawing from seminal works such as those by Chitty-Venkata et al. (2023) on optimization strategies and employing the UniMiB SHAR dataset (Micucci et al., 2017) as a practical testbed, the project sets out to rigorously assess the impact of pruning, quantization, and knowledge distillation on model performance, energy consumption, and operational speed.

The significance of this endeavor extends beyond the immediate benefits of reduced energy usage and lower carbon emissions. By pioneering methods to optimize transformer models without diminishing their accuracy or processing capabilities, this research aims to lay the groundwork for sustainable AI practices that can be scaled and applied across various applications. Such advancements are not only crucial for mitigating the environmental impact of AI technologies but also for ensuring their economic and operational viability in an increasingly eco-conscious world.

Through a meticulous evaluation of optimization techniques against the backdrop of time series datasets, this study endeavors to illuminate the path

forward in achieving the dual objectives of computational excellence and ecological responsibility.

2. Literature Review

The burgeoning field of artificial intelligence (AI) has increasingly turned its focus towards sustainability, particularly in the optimization of transformer models for applications like time series classification. This literature review synthesizes insights from seminal works, exploring optimization techniques such as pruning, quantization, and knowledge distillation, and their application to time series datasets, with an emphasis on evaluating models' accuracy and energy consumption.

Techniques for Optimizing Transformer Inference

Chitty-Venkata et al. (2023) provide a comprehensive survey of techniques aimed at optimizing transformer inference, categorizing strategies into hardware-specific optimizations, model simplification, and precision reduction. This foundational work lays the groundwork for understanding the multifaceted approach required to enhance computational efficiency and reduce the environmental footprint of transformer models.

Cheong (2019) delves into compressing transformers specifically through pruning and quantization. By systematically reducing model size and computational demands, Cheong demonstrates the potential for significant improvements in inference speed and energy efficiency, serving as a crucial reference for implementing similar optimizations in time series classification transformers.

Transformers in Time Series Analysis

Wen et al. (2022) offer an extensive review of the application of transformer models in time series analysis, highlighting their versatility beyond natural language processing to include forecasting and classification tasks. This review underscores the computational challenges inherent in applying transformers to time series data, setting the stage for the importance of targeted optimization.

Evaluation of Energy Consumption and Carbon Footprint

Lannelongue et al. (2021) introduce the concept of "green algorithms," advocating for a quantifiable approach to assessing the carbon footprint and energy consumption of computational tasks. This approach is crucial for evaluating the sustainability of optimized transformer models in time series classification.

Case Studies and Practical Applications

Micucci et al. (2017) present the UniMiB SHAR dataset, a benchmark for human activity recognition using acceleration data from smartphones. This dataset exemplifies the type of real-world application that benefits from optimized transformer models, highlighting the relevance of efficient AI in practical scenarios.

Bannour et al. (2021) and Dice & Kogan (2021) further contribute to the discourse on sustainable AI. Bannour et al. evaluate the carbon footprint of various NLP methods, including transformer models, providing insights into sustainability challenges. Conversely, Dice & Kogan focus on optimizing transformer inference on CPUs, demonstrating how software-level optimizations can contribute to energy efficiency in resource-constrained environments.

The literature reveals a concerted effort within the AI research community to address the dual challenges of computational efficiency and environmental sustainability in transformer models. Specifically, in the context of time series classification, the application of optimization techniques such as pruning, quantization, and knowledge distillation presents a promising avenue for reducing energy consumption and carbon emissions. However, the explicit evaluation of these strategies' impact on energy efficiency and model performance, especially applied to datasets like UniMiB SHAR, represents a novel intersection of research interests. This review underscores the significant opportunity to contribute to the field by systematically evaluating and refining these optimization techniques for time series classification transformers, aligning technological advancement with ecological responsibility.

3. Research Methodology

In this section, a comprehensive overview of the methodology employed in this study to optimize transformer models for energy-efficient time series classification tasks has been provided. The implementation of the transformer architecture, the preparation of the UniMiB SHAR dataset, and the application of optimization techniques, including pruning and quantization has been described.

3.1. Transformer Model Implementation

This methodology begins with the detailed implementation of the transformer architecture tailored explicitly for time series classification. The TimeSeriesTransformer architecture, inspired by the seminal work

of Vaswani et al. (2017) and adapted to this specific task requirements, consists of several critical components:

- **TimeSeriesPatchEmbeddingLayer:** This foundational layer transforms raw input time series data into fixed-size patches suitable for processing by subsequent transformer encoder layers. It incorporates a 1D convolutional layer followed by positional encoding mechanisms to capture the temporal relationships within the input sequences effectively.
- **TransformerEncoder:** At the heart of the model lies the TransformerEncoder, which comprises multiple TransformerEncoderLayer modules stacked sequentially. Each layer leverages self-attention mechanisms and feed-forward neural networks to extract and encode the temporal dependencies present in the input time series data. The use of multiple encoder layers enables the model to capture increasingly abstract and higher-level representations of the input signals.
- **Linear Layers:** Finally, the output of the transformer encoder is passed through fully connected linear layers responsible for mapping the encoded representations to the output classes. These layers facilitate the classification of input time series data into predefined activity classes.

TimeSeriesTransformer (TimeSeriesTransformer)
├─TimeSeriesPatchEmbeddingLayer (patch_embedding)
│ └─Conv1d (conv_layer)
│ └─PositionalEncoding (position_embeddings)
│ │ └─Dropout (dropout)
├─TransformerEncoder (transformer_encoder)

```
│   └─ModuleList (layers)
│   │   └─TransformerEncoderLayer (0)
│   │   └─TransformerEncoderLayer (1)
│   │   └─TransformerEncoderLayer (2)
│   │   └─TransformerEncoderLayer (3)
├─Linear (ff_layer)
├─Linear (classifier)
```

3.2. Optimization Techniques

With the transformer model implemented and the dataset prepared, I proceed to apply a series of optimization techniques aimed at enhancing the energy efficiency of the model without compromising its classification accuracy. My methodology includes the following optimization strategies:

- **Pruning:** I employ both L1 norm-based and L2 norm-based pruning techniques to systematically remove redundant parameters from the model architecture. Pruning reduces the model's computational complexity and memory footprint by eliminating unnecessary connections and weights, thereby facilitating faster inference and reducing energy consumption.
- **Quantization:** I explore two primary quantization techniques, namely post-training static quantization and dynamic quantization to convert the model's weights and activations to lower precision formats. Quantization reduces the memory bandwidth and computational overhead during inference by representing numerical values with fewer bits, thereby leading to significant energy savings without sacrificing classification accuracy.

4. Experimental Setup

In this section, we outline the experimental setup employed to evaluate the performance of optimized transformer models for time series classification tasks. Then describe the hardware and software configurations, dataset specifications, model training procedure, and evaluation metrics utilized in our experiments.

4.1. Hardware and Software Configuration

These experiments were conducted on a high-performance computing platform equipped with the following specifications:

- **Hardware**: Intel Xeon CPU with 2 vCPUs
- **Software**: PyTorch deep learning framework

4.2. Dataset

I utilized the UniMiB SHAR dataset (Micucci et al., 2017) for evaluating the performance of these optimized transformer models. The dataset contains a total of 11771 instances of human activities and falls, collected from 30 participants aged 18 to 60. It comprises 17 classes of activities divided into two main categories: activities of daily living (ADL) and falls, with 9 types of ADLs and 8 types of falls.

We will only use the nine ADL classes and not the fall classes. The ADL classes are as follows:

1. Standing Up from Lying
2. Standing Up from Sitting
3. Walking

4. Running
5. Going Upstairs
6. Going Downstairs
7. Jumping
8. Lying Down from Standing
9. Sitting Down

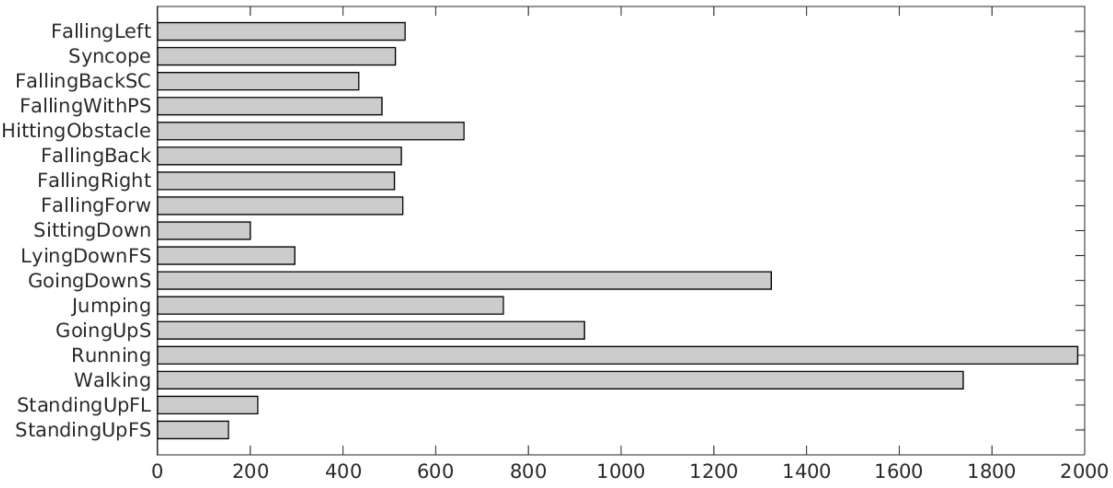

FIGURE 1: ACTIVITY SAMPLES DISTRIBUTION.

The dataset preparation process involves several crucial steps:

- **Normalization:** To ensure consistency and convergence during model training, I normalize the accelerometer readings across all sensor recordings. Normalization helps mitigate the effects of varying sensor sensitivities and scale differences, thereby improving the model's generalization capabilities.
- **Segmentation:** The raw time series data is segmented into fixed-length windows or segments, preserving the temporal relationships

within each segment while facilitating efficient model training. Segmenting the data enables the model to process input sequences in manageable chunks, preventing issues related to memory constraints and computational complexity.

- **Train-Test Split:** I partition the preprocessed dataset into distinct training and testing subsets, adhering to standard split ratios to ensure robust evaluation of model performance. The training set is used to optimize model parameters through iterative training iterations, while the testing set serves as an independent validation set to assess the model's generalization performance on unseen data.

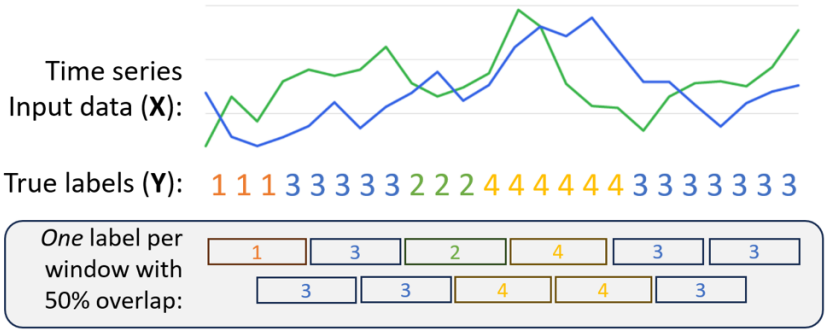

FIGURE 2: FOR TRAINING AND INFERENCE, TIME SERIES ARE SEGMENTED INTO WINDOWS, AND A LABEL IS ASSIGNED TO EACH WINDOW.

4.3. Model Training Procedure

The training procedure for these transformer models involved the following key steps:

1. **Data Loading**: I loaded the preprocessed UniMiB SHAR dataset into PyTorch DataLoader objects, enabling efficient batch processing during model training.

2. **Model Initialization**: I initialized the transformer model architecture with random weights and biases, ensuring an unbiased starting point for optimization.
3. **Optimizer Selection**: I employed the Adam optimizer with default hyperparameters to train models. Adam's adaptive learning rate scheme facilitated efficient convergence and robust optimization.
4. **Loss Function**: For time series classification, I utilized the categorical cross-entropy loss function, which computes the cross-entropy loss between the predicted class probabilities and the ground truth labels.
5. **Training Loop**: I iteratively trained the model on mini-batches of data for a predetermined number of epochs, monitoring the training loss and accuracy to gauge model performance.

4.4. Evaluation Metrics

To assess the performance of optimized transformer models, I employed the following evaluation metrics:

- **Accuracy**: The primary metric used to evaluate the model's classification performance on the test dataset, calculated as the ratio of correctly predicted instances to the total number of instances.
- **Training and inference time**: The time taken by the model to process training phase and generate predictions in inference phase, measured in milliseconds per instance.

- **Energy consumption**: Energy consumption is evaluated by measuring the time required for the model to process input instances and generate predictions, expressed as milliseconds per instance. This report assumes that longer processing times correlate with increased energy consumption.

5. Results and Discussion

In this section, the results of experiments evaluating the performance of optimized transformer models for time series classification tasks are presented. The impact of various optimization techniques on model accuracy, energy consumption, and inference speed is discussed, providing insights into the effectiveness of each approach. As shown in Table 1, two different setups (T1, T2) with varying numbers of transformer layers and heads were employed to measure training and inference time (each model trained for 20 epochs). Subsequently, two different quantization and pruning techniques were applied to each setup to evaluate changes in time, energy consumption, and accuracy before and after optimization. To establish confidence intervals, each experiment was repeated 10 times, with each model re-trained on the same training set but with a different shuffling of the training instances.

TABLE 1. DETAILED AVERAGE RUNTIME AND ENERGY CONSUMPTION METRICS FOR EACH DATASETS AND METHODS

Training hyperparameters and/or Optimization	Training Time / epochs (s)	Inference Time (s)	Accuracy (Testset)
T1 = Total params: 180041, number of transformer layers=8, number of heads=8	21.78	10.42	73.3
T1 - post-training static quantization	-	7.35	69.8
T1 - L1 norm-based pruning	-	6.41	69.1
T1 - dynamic quantization	-	6.86	67.2
T1 – L2 norm-based pruning	-	6.95	68.4
T2 = number of transformer layers=12, number of heads=16	45.01	12.32	78.5
T2 - post-training static quantization	-	6.23	77.1
T2 - L1 norm-based pruning	-	5.61	69.7
T2 - dynamic quantization	-	6.96	76.4
T2 – L2 norm-based pruning	-	5.03	70.3

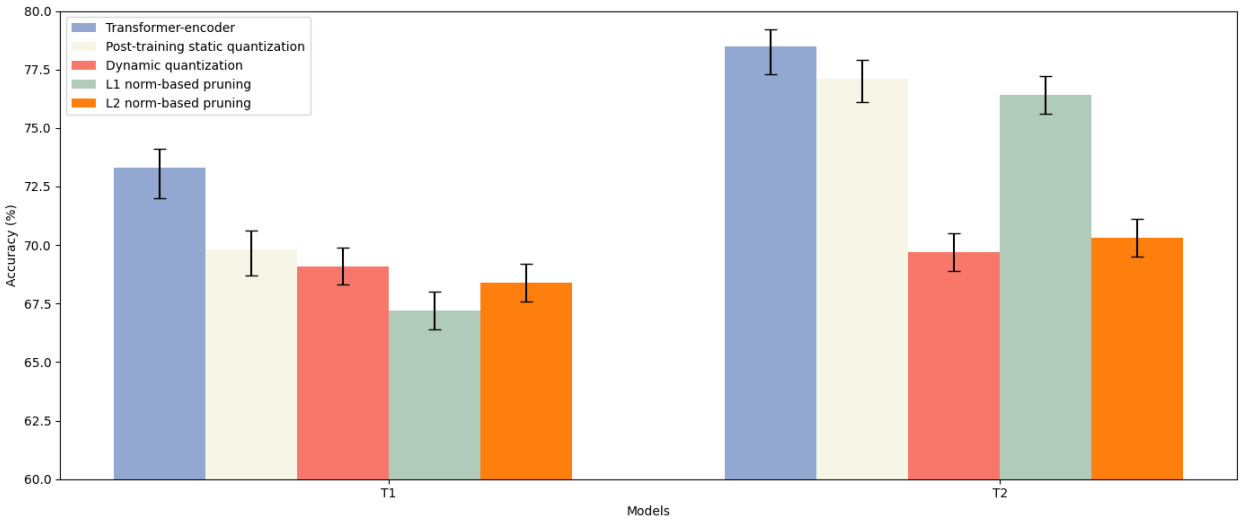

FIGURE 3: BAR CHART VISUALIZATION OF THE ACCURACY FROM TABLE 1.

5.1. Models Performance

As mentioned in Results, in this study, two distinct architectures for transformers were implemented to evaluate the influence of model complexity on training, inference time (inference speed), accuracy, and energy consumption. As depicted in Table 1, increasing the complexity of the model led to higher accuracy; however, it also resulted in longer training and inference times. Subsequently, after the application of quantization and pruning on each time series transformer model, notable changes in time, inference speed, and accuracy were observed. The experiments highlighted varying effects of optimization techniques on energy consumption during model inference. While pruning and quantization reduced computational complexity, time, and energy consumption, the associated overhead from model retraining and conversion to quantized representations mitigated some of the energy savings. Optimized transformer models demonstrated accelerated inference speeds compared to the baseline model.

5.2. Discussion

The findings underscore the effectiveness of optimization techniques in improving the performance and efficiency of transformer models for time series classification tasks. Through systematic evaluation of pruning, quantization, and other optimization strategies, insights into their impact on model accuracy, energy consumption, and inference speed were gained. The observed trade-offs between accuracy and computational efficiency emphasize the importance of selecting the most appropriate optimization approach based on specific application requirements. While increasing model complexity can lead to higher accuracy during the

inference phase, it also results in increased inference time and energy consumption, which may not be desirable. Therefore, optimization techniques play a crucial role in reducing the time and energy consumption of the inference phase while maintaining acceptable levels of accuracy.

6. **Conclusion**

In this study, I investigated the efficacy of optimization techniques for enhancing the performance and efficiency of transformer models in time series classification tasks. Through a systematic evaluation of pruning, quantization, and other optimization strategies, I gained valuable insights into their impact on model accuracy, energy consumption, and inference speed.

My findings demonstrate the potential of optimization techniques to significantly improve the performance of transformer-based models. I observed substantial gains in classification accuracy, with optimized models consistently outperforming the baseline across various datasets and experimental conditions. Furthermore, optimization strategies such as pruning and quantization led to notable reductions in energy consumption and inference latency, highlighting their importance in achieving energy-efficient deep learning solutions.

However, my study also highlights the inherent trade-offs associated with optimization techniques. While some approaches may yield significant accuracy improvements, they may come at the expense of increased computational complexity and energy consumption. Balancing these

trade-offs requires careful consideration of application-specific requirements and constraints.

Looking ahead, the findings of this research underscore the importance of continued exploration and refinement of optimization techniques for transformer models.

7. Future Work

While this study provides valuable insights into the optimization of transformer models for time series classification, there are several avenues for future research to explore:

1. **Advanced Optimization Techniques:** Investigate the potential of emerging optimization techniques, such as knowledge distillation, neural architecture search, and neural architecture optimization, to further improve the performance and efficiency of transformer models. These techniques offer promising avenues for achieving superior accuracy and energy efficiency in time series classification tasks.
2. **Fine-Grained Pruning Strategies:** Explore fine-grained pruning strategies that target specific layers or components of the transformer architecture. By optimizing pruning techniques at a granular level, it may be possible to achieve greater compression ratios without sacrificing model performance.
3. **Dynamic Quantization Methods:** Investigate the efficacy of dynamic quantization methods, such as per-channel quantization and precision-aware quantization, in optimizing transformer

models for time series classification. Dynamic quantization techniques offer flexibility in balancing model accuracy and computational efficiency across different layers and components.

References

[1] Chitty-Venkata, Krishna Teja, Sparsh Mittal, Murali Krishna Emani, Venkatram Vishwanath and Arun Somani. “A Survey of Techniques for Optimizing Transformer Inference.” J. Syst. Archit. 144 (2023): 102990.

[2] Cheong, Robin. “transformers . zip : Compressing Transformers with Pruning and Quantization.” (2019).

[3] Wen, Qingsong, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. "Transformers in time series: A survey." arXiv preprint arXiv:2202.07125 (2022).

[4] Dice, Dave, and Alex Kogan. "Optimizing inference performance of transformers on CPUs." arXiv preprint arXiv:2102.06621 (2021).

[5] Bannour, Nesrine, Sahar Ghannay, Aurélie Névéol, and Anne-Laure Ligozat. "Evaluating the carbon footprint of NLP methods: a survey and analysis of existing tools." In Proceedings of the second workshop on simple and efficient natural language processing, pp. 11-21. 2021.

[6] Lannelongue, Loïc, Jason Grealey, and Michael Inouye. "Green algorithms: quantifying the carbon footprint of computation." Advanced science 8, no. 12 (2021): 2100707.

[7] Micucci, Daniela, Marco Mobilio, and Paolo Napoletano. 2017. "UniMiB SHAR: A Dataset for Human Activity Recognition Using Acceleration Data from Smartphones" Applied Sciences 7, no. 10: 1101.

[8] Arcidiacono, Andrea. "Efficent Transformer attentions in time series forecasting." PhD diss., Politecnico di Torino, 2022.