ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF COMPUTER SCIENCE
AND ENGINEERING

ARTIFICIAL INTELLIGENCE

MASTER THESIS

in

Intelligent Systems

HARDWARE DIMENSIONING FOR ENVIRONMENTAL SUSTAINABILITY: BENCHMARK OF AI ALGORITHMS AND ENVIRONMENTAL IMPACT

CANDIDATE

Enrico Morselli

SUPERVISOR

Prof. Andrea Borghesi

CO-SUPERVISOR

Allegra De Filippo, PhD.

Academic year 2024-2025

Session 5th

dedicated(X) :- friend(X).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Rationale

In recent years we have witnessed a dramatic increase in the performance of Artificial Intelligence technologies. Even if AI still fails to exeed human ability in some complex cognitive tasks, as of 2023 it has surpassed human capabilites in a range of tasks, such as image classification, basic reading comprehension, visual reasoning and natural language inference [5]. Not to mention the astonishing results achieved by Generative AI in tasks as Image and Video Generation [5]. This great advances in performance were made possible by a massive upscale of model sizes and computational resources ("compute" in short) dedicated to training state-of-the-art AI models. Research shows that for frontier AI models (i.e. those that were in the top 10 of training compute when they were released), the training compute has grown by a factor of 4-5x/year since 2010 [8]. This surge in required compute has driven a corresponding spike in energy consumption for AI, and consequently, an higher environmental impact due to $\mathrm{CO_2}$ emissions. For instance, for training their LLaMA models, Meta AI researchers have estimated a period of approximately 5 months of on 2048 $\mathrm{A100}$ $\mathrm{80GB}$ GPUs, resulting in a total of $2,638\mathrm{MWh}$ of energy and a total emission of $1,015$ $\mathrm{tCO_2eq}$ [11]. Given the widespread application, the steep increase in model size and complexity and the crescent energy requirements for AI applications, the Carbon Footprint of AI has become a growing concern in the context of the current climate emergency.

In this work, we will explore an approach for addressing the issue of AI sustainability, by means of HADA (HArdware Dimensioning for AI Algorithms), which is a framework that uses ML to learn the relationship between an algorithm configuration and performance metrics, like total runtime, solution cost and memory usagem and then uses Optimization to find the best Hardware architecture and its configuration to run an algorithm under required performances and budget limits which is the problem known as Hardware Dimensioning [1]. What we will do is to extend this framework in order to consider also the performance of the algorithms in terms of Energy consumption and Carbon Emissions, so that, ideally, we could find the best Algorithm and Hardware configuration that reduces the Carbon Footprint of computation. We will then proceed to test this approach on some small-scale algorithms that we could easily execute in a timely manner on local machines and HPC clusters.

The rest of the work is structured as follows:

- **Chapter 2** Introduces Related works that addressed the issue of AI's carbon footprint, and how Carbon Footprint is determined
- **Chapter 3** Introduces some theoretical background about HADA, and explains the integration of the new metrics.
- **Chapter 4** Presents the experimental setup and the results of the experiments
- **Chapter 5** Presents the HADA framework, providing an overview of the tool
- **Chapter 6** Presents the conclusions

Chapter 2

Related Works

2.1 Sustainability in AI

The environmental impact of training large AI models is illustrated by recent empirical assessments. Strubell et al. (2019) quantified the CO2 emissions of several NLP models and found that training a big transformer with extensive hyperparameter tuning (including neural architecture search) emitted roughly 626,000 pounds of CO_2 - about the same as the lifetime emissions of five cars [9] [2]. These findings brought attention to the fact that accuracy gains in AI often come at a steep energy and carbon price. In response, researchers have begun to sistematically reporting energy use and consequent CO_2 emissions of model training to raise awareness [2] [6]. Beyond individual models, broader studies have examined AI's total energy and environmental footprint across the industry. Henderson et al. (2020) [4] introduced a framework for tracking real-time energy consumption and carbon emissions during ML experiments, encouraging researchers to include these metrics in publications. Their work underscored that transparent reporting is essential for understanding and ultimately reducing AI's climate impacts. Industry-scale analyses also reveal sobering trends. Gupta et al. (2021) [3] analyzed the end-to-end footprint of computing and found that while operational emissions (from running hardware) have been partly curbed by efficiency improvements, the overall carbon footprint of computing continues to grow due to increased scale. Notably, they showed that for modern data centers and mobile devices, manufacturing and infrastructure (embodied carbon) now account for the majority of emissions. In other words, as data centers adopt cleaner power, the emissions "hidden" in hardware supply chains (chip fabrication, server manufacturing, etc.) become a dominant concern. Similarly, Wu et al. (2022) [12] present a holistic study of AI at a large tech company (Meta/Facebook), examining the entire AI model lifecycle - from data processing and training to inference and hardware lifecycle. They report super-linear growth in AI workloads and infrastructure: for instance, daily inference operations doubled over a recent 3-year span, forcing a 2.5x expansion in server capacity. Crucially, Wu et al. also highlight that embodied carbon is an increasing fraction of AI's total footprint, echoing that improvements in hardware efficiency alone cannot eliminate AI's impact. Their analysis argues for looking beyond training alone - considering data center construction, supply chains, and the frequency of model retraining - to truly grasp AI's environmental impact.

A recurring theme in these studies is the diminishing return on energy investment for AI model improvements. As models get larger and more complex, the incremental accuracy gains often require disproportionately more compute (and thus energy). Schwartz et al. (2020) [7] dub the status quo "Red AI," where researchers prioritize accuracy at almost any computational cost, and they note this trend is unsustainable both environmentally and even economically. Thompson et al. (2021) [10] similarly observed that progress in benchmarks was coming with exponentially increasing computing cost, warning of diminishing returns and calling the situation unsustainable. Another challenge is equitable access: massive energy requirements make cutting-edge AI research expensive, potentially concentrating it in wealthy institutions and regions with robust infrastructure. This raises concerns that AI's growing energy hunger not only harms the planet but also exacerbates inequalities in who can afford to do top-tier research. These concerns have prompted calls for a paradigm shift toward "Green AI", where efficiency and sustainability are treated as primary goals in model development. Several studies have addressed the issue of AI's carbon footprint.

2.2 Tools for tracking Carbon Emissions

The growing awareness about AI's Carbon Footprint also motivated the development of dedicated tools and methodologies to monitor the carbon footprint of AI workloads. A number of open-source tools and frameworks have been created to help practitioners measure the energy consumption and CO_2 emissions of their code. Among those tools, there is **CodeCarbon**, which is the tool we used to expand HADA in this work. CodeCarbon is an open-source Python package for tracking the carbon footprint of computing projects. It integrates into ML code to log the resources used (CPU, GPU, etc.) and estimates the CO_2 emissions produced by the workload. Uniquely, CodeCarbon accounts for the location of the computation - using region-specific electricity carbon intensity data - to provide location-dependent emission estimates. This allows developers to see, for instance, that running the same training job on a low-carbon grid (e.g. hydro-powered Montreal) results in a much smaller footprint than running it on a coal-heavy grid. The goal is to inform and incentivize researchers and engineers to optimize or relocate their workloads to reduce emissions.

Tools like **Green Algorithms** and **CodeCarbon** have been developed to estimate and monitor emissions.

CodeCarbon is an open-source tool designed to track the energy consumption of computational resources and estimate the corresponding carbon emissions. The formula used is:

$$CO2eq = C \times E \tag{2.1}$$

where:

- **C** represents the carbon intensity of electricity (kg CO2e per kWh), varying by country and energy mix.
- **E** is the total electricity consumed during computation (kWh).

By monitoring CPU, GPU, and RAM consumption, CodeCarbon estimates the total emissions associated with a computation. It retrieves the carbon intensity based on the geographical location and logs results at user-defined intervals (default: 15 seconds).

Installation:

```
pip install codecarbon
```

Chapter 3

Metodology

3.1 Empirical Model Learning in HADA

HADA employs the **Empirical Model Learning (EML)** paradigm, which integrates **Machine Learning (ML)** models into an optimization framework. EML involves:

1. **Data Collection**: Running target algorithms under various hyperparameter configurations and hardware settings to collect performance data.

2. **Surrogate Model Creation**: Training ML models (e.g., Decision Trees) to approximate the relationship between input configurations and performance metrics (e.g., runtime, memory, carbon emissions).

3. **Optimization**: Using the learned models within a combinatorial optimization framework to find the best hardware configuration.

HADA was originally applied to the **ANTICIPATE** and **CONTINGENCY** stochastic algorithms used in energy management. These algorithms compute energy production schedules while minimizing cost and considering uncertainties.

3.2 Integration of CodeCarbon in HADA

To extend HADA for sustainable AI, we integrate CodeCarbon to track emissions in:

- ANTICIPATE and CONTINGENCY algorithms.
- MaxFlow Algorithms:
  - Boykov-Kolmogorov (BK)
  - Excess Incremental Breadth First Search (EIBFS)
  - Hochbaum's Pseudo Flow (HPF)

Chapter 4

Experimental Analysis

4.1 Benchmarking on Different Hardware Platforms

Experiments were conducted on:

- MacBook Pro (2019)
- Leonardo Supercomputer (CINECA HPC)

Each algorithm was executed on 30 instances with hyperparameter values ranging from 1 to 100, generating datasets with 6,000 records per algorithm per hardware platform.

Chapter 5

HADA-as-a-Service

5.1 HADA Web Application

Benchmark data was integrated into the HADA web application, requiring:

- Creation of JSON configuration files for each algorithm-hardware combination.
- Specification of hyperparameters and performance targets.

Example JSON structure:


```
{
"name": "anticipate",
"HW_ID": "macbook",
"hyperparams": [
{"ID": "num_scenarios", "type": "int", "LB": 1, "UB": 100}
],
"targets": [
{"ID": "time", "LB": null, "UB": null},
{"ID": "memory", "LB": null, "UB": null},
{"ID": "emissions", "LB": null, "UB": null}
]
```

```
}
```

Chapter 6

Conclusions

This work extends HADA by integrating carbon emission constraints, enhancing its applicability for sustainable AI hardware selection. Through experimental benchmarks on laptops and HPC systems, we validated the framework's ability to balance performance and environmental impact. The web-based prototype enables users to make informed decisions when configuring AI workloads under sustainability constraints.

Bibliography

[1] A. De Filippo et al. "HADA: An automated tool for hardware dimensioning of AI applications". In: *Knowledge-Based Systems* 251 (2022), p. 109199. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2022.109199. URL: https://www.sciencedirect.com/science/article/pii/S0950705122005974.

[2] J. Dodge et al. "Measuring the Carbon Intensity of AI in Cloud Instances". In: *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '22. Seoul, Republic of Korea: Association for Computing Machinery, 2022, pp. 1877–1894. ISBN: 9781450393522. URL: https://doi.org/10.1145/3531146.3533234.

[3] U. Gupta et al. "Chasing Carbon: The Elusive Environmental Footprint of Computing". In: *CoRR* abs/2011.02839 (2020). arXiv: 2011.02839. URL: https://arxiv.org/abs/2011.02839.

[4] P. Henderson et al. "Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning". In: *Journal of Machine Learning Research* 21.248 (2020), pp. 1–43. URL: http://jmlr.org/papers/v21/20-312.html.

[5] N. Maslej et al. *The AI Index 2024 Annual Report*. Tech. rep. AI Index Steering Committee, Stanford University, 2024.

[6] D. A. Patterson et al. “Carbon Emissions and Large Neural Network Training”. In: *CoRR* abs/2104.10350 (2021). arXiv: 2104.10350. URL: https://arxiv.org/abs/2104.10350.

[7] R. Schwartz et al. “Green AI”. In: *CoRR* abs/1907.10597 (2019). arXiv: 1907.10597. URL: http://arxiv.org/abs/1907.10597.

[8] J. Sevilla and E. Roldán. *Training Compute of Frontier AI Models Grows by 4-5x per Year*. Accessed: 2025-03-03. 2024. URL: https://epoch.ai/blog/training-compute-of-frontier-ai-models-grows-by-4-5x-per-year.

[9] E. Strubell, A. Ganesh, and A. McCallum. “Energy and Policy Considerations for Deep Learning in NLP”. In: *CoRR* abs/1906.02243 (2019). arXiv: 1906.02243. URL: http://arxiv.org/abs/1906.02243.

[10] N. Thompson et al. “Deep Learning’s Diminishing Returns: The Cost of Improvement is Becoming Unsustainable”. In: *IEEE Spectrum* 58 (2021), pp. 50–55. ISSN: 0018-9235.

[11] H. Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL]. URL: https://arxiv.org/abs/2302.13971.

[12] C. Wu et al. “Sustainable AI: Environmental Implications, Challenges and Opportunities”. In: *CoRR* abs/2111.00364 (2021). arXiv: 2111.00364. URL: https://arxiv.org/abs/2111.00364.

Acknowledgements

I'm very grateful to the inventor of the Prolog language, without whom this thesis couldn't exist. I'd also like to acknowledge my advisor Prof. Mario Rossi by tail-recursively acknowledging my advisor.