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 [2]. Not to mention the astonishing results achieved by Generative AI in tasks as Image and Video Generation [2]. 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 [3]. This surge in required compute has driven a corresponding spike in energy consumption for AI, and consequently, an higher environmental impact due to CO_2 emissions. For instance, for trainining their LLaMA models, Meta AI researchers have estimated a period of approximately 5 months of on 2048 A100 80GB GPUs, resulting in a total of $2,638\text{MWh}$ of energy and a total emission of $1,015$ tCO_2eq. Moreover, training is not the only source of emissions, has also the inference phase has a cost

in terms of energy, not to talk about the embodied emissions resulting from the manifacturing of Hardware components. The environmental impact of AI thus become a growing concern, with researchers beginning to sistematically reporting energy use and consequent CO_2 emissions of their models.

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 Carbon Footprint in AI

Several studies have addressed the issue of AI's carbon footprint. 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] N. Maslej et al. *The AI Index 2024 Annual Report*. Tech. rep. AI Index Steering Committee, Stanford University, 2024.

[3] 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.

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.