Lifecycle Assessment of a Machine Learning Algorithm: A Case Study

Victor Charpenay[1,2], Audrey Tanguy[1,3]

[1] Mines Saint-Étienne
[2] Laboratoire d'informatique, de modélisation et d'optimisation des systèmes (LIMOS)
[3] Laboratoire Environnement, ville et société (EVS)

28 March 2023

Context & Objective

Institut Henri Fayol is a **multidisciplinary research** center of Mines Saint-Étienne. It hosts researchers in the domains of:

- mathematics and data science
- computer science
- environmental science
- management

Context & Objective

Institut Henri Fayol is a **multidisciplinary research** center of Mines Saint-Étienne. It hosts researchers in the domains of:

- mathematics and data science
- computer science
- environmental science
- management

Studying the environmental impact of computing is often at the intersection of (at least two of) these four domains.

Context & Objective

The present work is at the intersection of data science and environmental science.

Context & Objective

The present work is at the intersection of data science and environmental science.

Its objective is to apply proven **Lifecycle Assessment** (LCA) methods to a **Machine Learning** (ML) service.

Learning vs. Usage

Recent advances in ML come at the cost of a significant **increase in computation**.

Learning vs. Usage

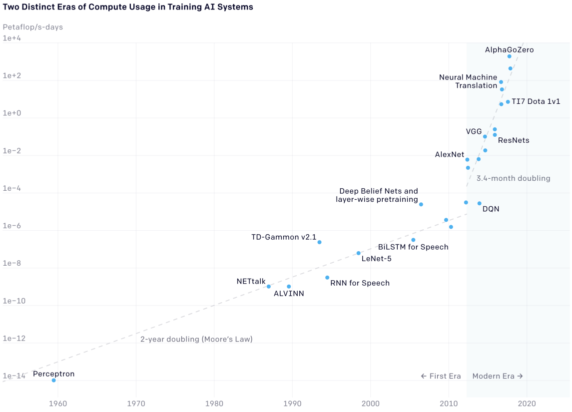


Figure: Increase in ML algorithm computation over years (OpenAI, 2018)

Learning vs. Usage

Current research focuses on **minimizing emissions** induced by **training** ML models.

Learning vs. Usage

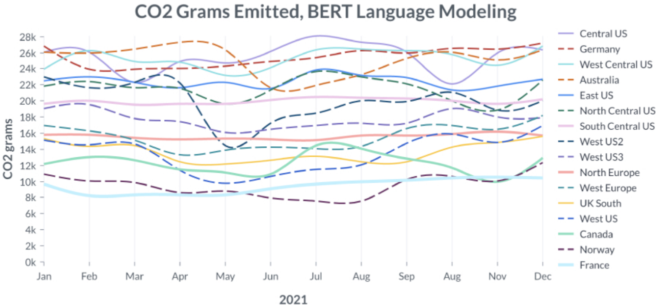


Figure: Seasonal variations in emissions for training the BERT large language model (Dodge *et al.*, 2022)

Learning vs. Usage

Yet, the learnt models, especially large language models, can easily be **shared** and used with **little to no retraining**.

Learning vs. Usage

Yet, the learnt models, especially large language models, can easily be **shared** and used with **little to no retraining**.

ChatGPT required **more than 1,000 petaflop/s-days** to train but **100+ million persons** used it in January 2023. Flops “per capita” are low.

Learning vs. Usage

Yet, the learnt models, especially large language models, can easily be **shared** and used with **little to no retraining**.

ChatGPT required **more than 1,000 petaflop/s-days** to train but **100+ million persons** used it in January 2023. Flops "per capita" are low.

Does a question submitted to ChatGPT emit more than a light bulb turned on for 1h?

Learning vs. Usage

*Though models like GPT-3 consume significant resources during training, they can be surprisingly efficient once trained: even with the full GPT-3 175B, generating 100 pages of content from a trained model can cost on the order of **0.4 kW-hr** (Brown et al., 2020)*

Learning vs. Usage

> *Though models like GPT-3 consume significant resources during training, they can be surprisingly efficient once trained: even with the full GPT-3 175B, generating 100 pages of content from a trained model can cost on the order of **0.4 kW-hr** (Brown et al., 2020)*

Still, during inference, a (one-page long) answer given by GPT-3 would consume as much as **20 min of CPU activity** (e.g. to query a large database).

Learning vs. Usage

*Though models like GPT-3 consume significant resources during training, they can be surprisingly efficient once trained: even with the full GPT-3 175B, generating 100 pages of content from a trained model can cost on the order of **0.4 kW-hr** (Brown et al., 2020)*

Still, during inference, a (one-page long) answer given by GPT-3 would consume as much as **20 min of CPU activity** (e.g. to query a large database).

Over its entire lifecycle, would **ChatGPT** consume more than **Wikipedia**?

Case Study

We don't have the computational resources to experiment with ChatGPT and/or Wikipedia.

Case Study

We don’t have the computational resources to experiment with ChatGPT and/or Wikipedia.

But we have contact with companies providing more standard Machine Learning services.

Case Study

OpenStudio, a software development company, provides Machine Learning services on top of their e-commerce platform (Thelia).

Case Study

OpenStudio, a software development company, provides Machine Learning services on top of their e-commerce platform (Thelia).

Our case study is a **recommender system** trained over user interactions and product features.

System Architecture

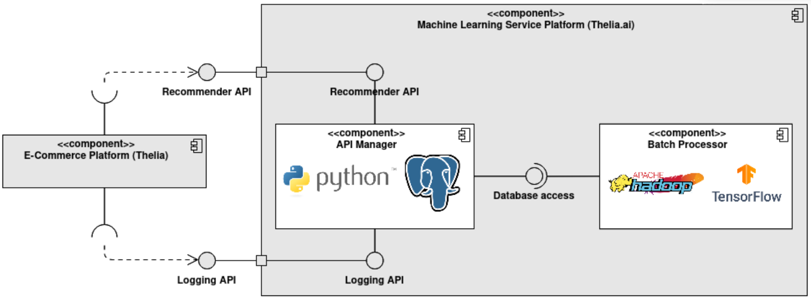


Figure: Architecture of the Thelia.ai platform

System Architecture

The two service components (API manager, batch processor) are each deployed in a **Docker container**.

System Architecture

The two service components (API manager, batch processor) are each deployed in a **Docker container**.

The entire ML service platform is hosted on a **Virtual Private Server** (VPS).

System Frontiers

Application logs weren't provided by OpenStudio.

System Frontiers

Application logs weren’t provided by OpenStudio.

The analyzed system thus reduces to the **batch processing component**.

System Frontiers

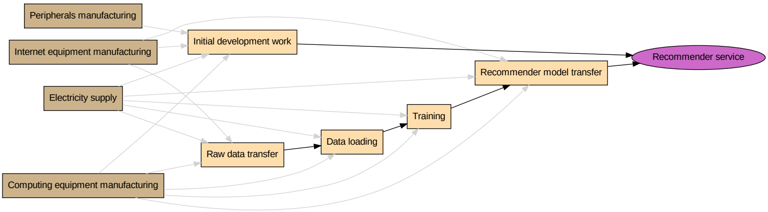


Figure: Processes involved in the development and operations of a Machine Learning service and their dependencies

Assumptions

The following figures were provided by OpenStudio:

Assumptions

The following figures were provided by OpenStudio:

- 2500 h of initial development work were needed
  - 5 persons worked over 9 months on the project

Assumptions

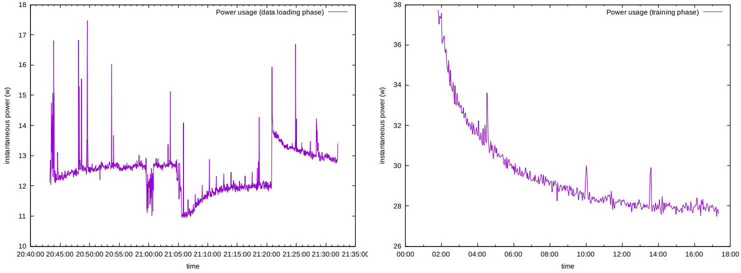
The following figures were provided by OpenStudio:

- 2500 h of initial development work were needed
  - 5 persons worked over 9 months on the project
- a recommender model is trained daily
  - transfered data (for 1 day) is $<$ 50 MB
  - training is over 36 months of data (7 GB)
  - loading data takes 45 min
  - training takes 15 min

Assumptions

We **extrapolated energy consumption** from power measurements on a standard TensorFlow model for recommender systems, applied to a large benchmark (MovieLens 20M).

Assumptions



(a) Data loading (b) Training

Figure: Instantaneous power as measured by HWInfo

Env. Impact Results

Our overall carbon impact estimate of Thelia.ai's service, assumed to run over 2.5 years, is **63.30 kgCO2e**.

Env. Impact Results

Process	**GWP100 (kgCO2eq)**
Init. dev. work	57.02
Raw data transfer	0.0033
Data loading	4.44
Training	1.83
Recommender model transfer	0.0033

Table: Global warming power over 100 years (GWP100) per process in the service's lifecycle

Env. Impact Results

The carbon impact of **initial development effort** on Thelia.ai amounts for 90% of the total impact.

Env. Impact Results

The carbon impact of **initial development effort** on Thelia.ai amounts for 90% of the total impact.

Data loading only amounts for 70% of the impact during service operations.

Discussion

Results for the **operations phase** is of the same order of magnitude as other calculation methods.

Calculation method	**GWP100 (kgCO2e)**
ours	6.27
Green Algorithms	3.40
ML CO2 Impact	1.92

Table: Carbon impact for processes taking place during service operations

Discussion

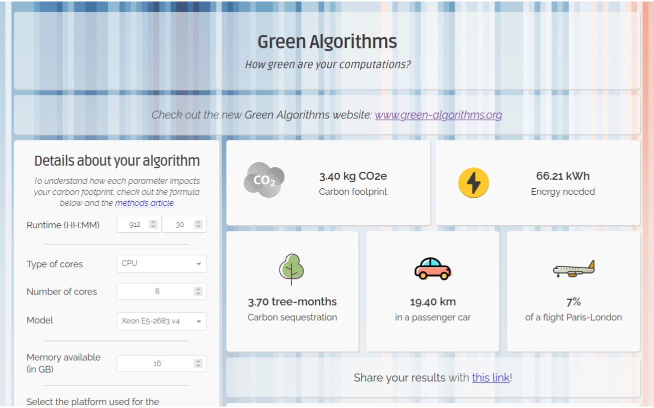


Figure: Carbon impact of Thelia.ai as given on `green-algorithms.org`

Discussion

Yet, our modeling of Cloud compute resources is optimistic:

- virtualization's overhead is considered to be 0
- compute load factor is assumed to be 1

Discussion

Yet, our modeling of Cloud compute resources is optimistic:

- virtualization’s overhead is considered to be 0
- compute load factor is assumed to be 1

In reality, Cloud servers are **not all active at all times.**

Discussion

Further, our calculation ignores API calls, which may have a significant impact on the compute load factor.

Discussion

Further, our calculation ignores API calls, which may have a significant impact on the compute load factor.

Thelia.ai's service receives $\sim$**15k Web requests per day**:

1. if responding takes 100% of the server's remaining resources, energy consumption is multiplied by 24;
2. if a request generates 1s of computation, energy consumption is multiplied by 5.

Discussion

Further, our calculation ignores API calls, which may have a significant impact on the compute load factor.

Thelia.ai's service receives $\sim$**15k Web requests per day**:

1. if responding takes 100% of the server's remaining resources, energy consumption is multiplied by 24;
2. if a request generates 1s of computation, energy consumption is multiplied by 5.

(These are probably overestimates...)

Conclusion

Measuring the impact of **standard** Machine Learning systems should include:

- initial development effort
- data processing
- data collection

Conclusion

Measuring the impact of **standard** Machine Learning systems should include:

- initial development effort
- data processing
- data collection

(*In our case study, data processing can probably be optimized.*)

Conclusion

The carbon impact of such systems can be reduced via:

- “models off the shelf” (*or no model at all?*)
- long-term support of Machine Learning systems ($>$ 5 years)