How can we estimate the energy consumption of training an AI model?

Presentation and comparison of existing tools
Why?

• Digital technologies emitted 3.5% of greenhouse gas emissions in 2019

• Its impact grows 6% by year

Figure 1: Évolution 2013-2025 de la part du numérique dans la consommation d'énergie primaire mondiale
(The Shift Project – Forecast Model 2021)
Energy consumption
Estimation

From hardware characteristics
(Thermal design power)

Energy consumption
Estimation

From hardware characteristics (Thermal design power)
Energy consumption

Estimation
From hardware characteristics
(Thermal design power)

External measures
Power measures from outside the hardware
Energy consumption

Estimation
From hardware characteristics
(Thermal design power)

Software power models
From hardware performance counters

External measures
Power measures from outside the hardware
Python packages

```python
import impactlib
# Loading dataset and processing it
tracker = impactlib.init()
tracker.start()
# Training
tracker.stop()
# Exporting results:
tracker.energy_consumption
tracker.carbon_emissions
```

Software power models
From hardware performance counters

Profiling softwares
Taking into account external factors

\[ \text{Energy}_{training} = \text{Energy} \times \text{External Factors} \]

- With External Factors:
  - **PUE**: Power Usage Effectiveness (To take into account everything that is necessary for data centers to run, like cooling)
  - **PSF**: Pragmatic Scaling Factor (To take into account hyper parameter search)
From energy to carbon emissions

- **Energy** kWh
- **Carbon emissions** gCO2 eq
- **Carbon intensity** gCO2eq/kWh

\[ \text{Emissions}_{\text{carbon}} = \text{Energy}_{\text{training}} \times \text{Intensity}_{\text{carbon}} \]
Experimental protocol

• **Application**
  - Image classification: Handwritten digit recognition
  - Model CNN: 2 convolutional layers, 2 fully connected layers (1 199 882 parameters, 5Mb)
  - Dataset: MNIST
  - Trained during 5 epochs (~ 120 seconds)
  - Library: Pytorch

• **Hardware**: gemini cluster (Grid’5000)
  - CPU: Intel Xeon E5-2698 v4 (Broadwell, 2.20GHz, 2 CPUs/node, 20 cores/CPU)
  - GPU: 8 x Nvidia Tesla V100-SXM2-32GB (32 GiB)

• **External factors**
  - Expected PUE: 1.58
  - Expected Carbon Intensity: 50 gCO2/kWh
Comparison of tools

• **Hardware estimations**
  - Green algorithms
  - ML CO2 Impact

• **External measures**
  - Wattmeters

• **Based on software power models**
  - CodeCarbon
  - Experiment-impact-tracker
  - CarbonTracker
  - Energy Scope
Comparison of all selected tools

Table 4: Energy and carbon emission parameter values

<table>
<thead>
<tr>
<th></th>
<th>GA</th>
<th>MCI</th>
<th>W</th>
<th>CC</th>
<th>EIT</th>
<th>CT</th>
<th>ES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carbon intensity (gCO2/kWh)</td>
<td></td>
<td></td>
<td></td>
<td>242.6</td>
<td>47.60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PUE</td>
<td>1.67</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>1.58</td>
<td>1.58</td>
<td>X</td>
</tr>
</tbody>
</table>

Figure 6: Energy: Given versus Expected

Figure 7: Carbon Emissions: Given versus Expected
Evaluation

- If I had run this during 24h
- Energy = 116.82 kWh

You could boil more than 1000L of water with this energy.

- Carbon footprint = 4 550 gCO2 eq

Driving 26 km by car emits the same carbon emissions.
Conclusion

• Various tools to compute energy consumption of training an AI model

• Provide it when publishing your model!

• Limits
  • Life Cycle Assessment
  • Cost of production and transport
  • Inference cost and the usage of the model
How to reduce this impact

• Be careful of which Data Center (PUE) you use, and of the location of it (carbon intensity) [GA][CC][EIT][CT]

• Algorithm optimization (random search, memory) [GA][CC][CT]

• Report energy metric and make energy efficient configurations more accessible [CC][EIT]

• Run your algorithms when carbon intensity is low [CC][CT]

• Adapt your algorithm and the settings to the hardware [CC][CT]

• Use pre-trained models [CC]

• Offset emissions [GA]
References

- ECOINFO: https://ecoinfo.cnrs.fr/
- GREEN ALGORITHMS [GA]
  - https://green-algorithms.org/?runTime_hour=24&runTime_min=0&locationContinent=Europe&locationCountry=France&locationRegion=FR&PUEradio=Yes&PUE=1.67&coreType=Both&numberCPUs=40&CPUmodel=Xeon%20E5-2697%20v4&usageCPUradio=Yes&usageCPU=1&numberGPUs=8&GPUmodel=NVIDIA%20Tesla%20V100&usageGPUradio=Yes&usageGPU=1&memory=512&platformType=localServer&PSFradio=Yes&PSF=1
- ML CO2 IMPACT [MCI]: https://mlco2.github.io/impact/?#compute
- CODECARBON [CC]
  - https://github.com/mlco2/codecarbon
  - medium article: https://medium.com/bcggamma/ai-computing-emits-co2%82%82-we-started-measuring-how-much-807dec8c35e3
References

• CARBONTRACKER [CT]
  • https://github.com/lfwa/carbontracker

• EXPERIMENT IMPACT TRACKER [EIT]
  • https://github.com/Breakend/experiment-impact-tracker

• ENERGY SCOPE [ES]
  • https://sed-bso.gitlabpages.inria.fr/datacenter/energy_scope.html (to request code)
  • http://energy-scope.bordeaux.inria.fr/ (INRIA server)

• Sources for carbon intensity
  • https://www.epa.gov/egrid/egrid-summary-tables
  • https://ourworldindata.org/grapher/co2-per-unit-energy
  • https://electricitymap.org/
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>By who</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Technical</td>
<td>Availability of source code</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Online availability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Easiness of installation and use</td>
<td>No install needed</td>
<td>No install needed</td>
<td>Poor</td>
<td>Quite good</td>
<td>Quite good</td>
<td>Good</td>
</tr>
<tr>
<td></td>
<td>Quality of documentation</td>
<td>Fair</td>
<td>Fair</td>
<td>Good</td>
<td>Fair</td>
<td>Good</td>
<td>Good</td>
</tr>
<tr>
<td></td>
<td>Output formats</td>
<td>Graphs and comparisons</td>
<td>Texts and Latex code</td>
<td>JSON</td>
<td>CSV file - Visualisation with Comet</td>
<td>JSON file - Code interface</td>
<td>Logs - Code interface</td>
</tr>
<tr>
<td>Configuration</td>
<td>Provide carbon emissions</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Carbon intensity source</td>
<td>Local, from electricityMap (2020 value)</td>
<td>Local, listed, including electricitymap</td>
<td>Local, global default value (didn't find source)</td>
<td>Local, from electricityMap (2018 value)</td>
<td>Various sources, European default value (EU-28 2017)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Carbon intensity value for my experiments (gCO2eq/kWh)</td>
<td>39</td>
<td>432</td>
<td>424.6</td>
<td>47.6</td>
<td>294.21</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Possible manual configuration of carbon intensity</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes but not easy</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>External factors taken into account (PUE, PSF)</td>
<td>PUE, PSF</td>
<td>Only for cloud providers</td>
<td>None</td>
<td>PUE</td>
<td>PUE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Carbon intensity value for my experiments (gCO2eq/kWh)</td>
<td>2019 value</td>
<td>2018 value</td>
<td>1.07</td>
<td>1.58</td>
<td>1.67</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Possible manual configuration of external factors</td>
<td>Yes, default value being PUE = 1.07 (2019), PSF = 1</td>
<td>No</td>
<td>No</td>
<td>Yes, default value PUE = 1.58</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Type of equipement</td>
<td>PC, server, Cloud</td>
<td>PC, server, Cloud</td>
<td>PC, server, Cloud</td>
<td>PC, server, Cloud</td>
<td>PC, server, Cloud</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Other</td>
<td>conversion to other metrics</td>
<td>conversion to other metrics</td>
<td>Interval between measures is configurable, conversion to other metrics, if tracking tools are unavailable will use TDPs from detected hardware, if hardware unknown will provide a default value</td>
<td>Interval between measures is configurable, predictions after one epoch, conversion to other metrics</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Energy model</td>
<td>Hardware taken into account</td>
<td>CPUs, GPUs, Memory</td>
<td>CPUs, GPUs, Memory</td>
<td>CPUs, GPUs, Memory</td>
<td>CPUs, GPUs, Memory</td>
<td>CPUs, GPUs, Memory</td>
</tr>
<tr>
<td></td>
<td>Model used</td>
<td>maximal usage power</td>
<td>maximal usage power</td>
<td>Measure</td>
<td>RAPL files - NVIDIA SMI</td>
<td>RAPL files - NVIDIA SMI</td>
<td>RAPL files - NVIDIA SMI</td>
</tr>
<tr>
<td></td>
<td>Estimation or Measure</td>
<td>Estimation from maximal usage power</td>
<td>Estimation from maximal usage power</td>
<td>Measure</td>
<td>Estimation from performance counters</td>
<td>Estimation from performance counters</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Real time</td>
<td>No</td>
<td>No</td>
<td>1/2</td>
<td>Yes</td>
<td>No</td>
<td>Per epoch</td>
</tr>
</tbody>
</table>