# Capability and Applicability of Measurement Tools for AI Model's Environmental Impact

Rui Zhou, Tao Zheng, Xin Wang, Lan Wang
Orange Innovation China
Beijing, China
e-mail: {rui.zhou, tao.zheng, xin2.wang, lan.wang}@orange.com
Emilie Sirvent-Hien
Orange Innovation
Châtillon, France
e-mail: emilie.hien@orange.com

# Topics of research interest of our workgroup

## Objectives:  AI carbon footprint/environmental impact study

- **Understand the environmental impact of AI models with measurement tools and monitoring tools**
- **Challenge AI use case design including edge computing regarding environmental impact**
- **Build recommendation on AI use case design**
- **Develop new measurement tools for AI use cases**

## Outline

- Analysis of measurement tools
- Case study: Re-ID
- Conclusions & future work

# AI models & measurement and monitoring tools evaluation

| Family | Tools | Use | Evaluation |
|---|---|---|---|
| A Priori measurement | keras-flops | These tools are used to compute the AI model's FLOPs/Mult-Adds and other related measurements to evaluate algorithms.<br><br>Calculate the number of mathematical operations needed for training and inference | AI use cases:<br>• Object detection<br>• ReID<br>• Fashion detection<br>• Pose detection<br>etc. |
| | torchsummaryX | | |
| | torchstat | | |
| | flops-counter | | |
| On the fly measurement | Carbonai | Measure electricity consumption during computation | |
| | Power API (JouleHunter, PyJoules) | Monitoring hosting infrastructures, compatible with monitoring dashboards | |
| | jtop | Global consumption of all processes running on the machine with GUI | heterogenous physical infrastructure:<br>• intel NUC<br>• Nvidia Jetson Xavier, TX2, Nano<br>• Raspberry pi4 pi3<br>etc. |
| A Posteriori measurement | MLCO2 Impact | Estimate CO2 eq.  linked to a given computation | |
| | Green Algorithms | Estimate CO2 eq. linked to a given computation | |

# Measurement tool analysis – A Priori Measurement Tools summary

- **The effectiveness of priori measurement tools relies on their detailed implementation.**
- **The application of priori measurement tools are limited.**
  - The tools we tested just supports one special framework (tensorflow or PyTorch) and a subset of types of model layer
  - In practice, most of AI models usually include some specific layers (e.g., 3D ConvTranspose layer) which can not be calculated by our tested priori measuring tools.

| tools | support framework | outputs |
| --- | --- | --- |
| keras-flops | Tensorflow | flops |
| torchsummaryX | PyTorch | flops, Multi-Add, memory, total params |
| torchstat | PyTorch | Multi-Add, total params |
| flops-counter | PyTorch | Multi-Add, total params |

# Measurement tool analysis – on the fly measurement tools summary

- All those tools are easy to install and use, some of them can be installed with several command lines, like JouleHunter, CarbonAI and Jtop; and for PyJoules, we can add it into our application code just like a function.
- For compatibility, except CarbonAI support Linux, windows and MacOS (we only using it on Linux machines), other tools currently can only be used on Linux.
- Most On the fly tools can directly get power consumption of most components (CPU, GPU, RAM) targe program but only support Python. And for Jtop, worked well with ARM devices but it only works with Nvidia Jetson platforms, it has a not bad graphical.

| tools | support architecture | support OS | Support components | Usages |
|---|---|---|---|---|
| PyJoules | x86 | Linux | Duration, Intel CPU, Nvidia GPU,RAM | Python package |
| JouleHunter | x86 | Linux | Duration, Intel CPU, RAM | Python program |
| CarbonAI | x86 | Linux, windows and MacOS | Duration, Intel CPU, GPU,RAM | Python package, csv file |
| Jtop | arm | Linux | Nvidia Jetson platforms | Visualization tool |

# Measurement tool analysis - a posteriori measurement tools summary

- **Both tools use Thermal Design Power and runtime to calculate the power consumption, which means the usage of cores is 100% by default.** Green Algorithms takes more quantifiable elements into consideration, i.e., memory power, the real usage of cores, PUE, PSF, allowing users to estimate the power consumption more flexibly.

| | ML CO2 Impact | Green Algorithms |
|---|---|---|
| **Energy consumption** | **runtime * power draw for GPU** $E = t \times P_c$ | **runtime * (power draw for cores * usage + power draw for memory) * PUE * PSF** $E = t \times (n_c \times P_c \times u_c + n_m \times P_m) \times PUE \times PSF$ |
| **Hardware type** | • **Mainly GPU type** | • **GPU, CPU, CPU/GPU co-existing case, number of cores, memory** |
| **Usage factor** | • **100% by default** | • **100% by default and configurable** |
| **Other factors** | **/** | • **Power Usage Effectiveness: the extra energy needed to operate the data center (cooling, lighting etc.)**<br>• **Pragmatic Scaling Factor: multiple identical runs (e.g. for testing or optimization)** |

# Measurement tool analysis - a posteriori measurement tools summary

- **Both tools calculate carbon emission based on power consumption and carbon intensity of local grids.** For carbon intensity, there are various data sources that may provide quite different location scope and effective time, and ML $CO_2$ Impact covers more data compared with the other tool.

| | ML CO2 Impact | Green Algorithms |
|---|---|---|
| **Carbon intensity** | A measure of how much $CO_{2e}$ emissions are produced per kilowatt hour of electricity consumed. | |
| **Data centers** | • Google Cloud Platform, Amazon Web Services, Azure, OVHCloud, and Scaleway | • Google Cloud Platform, Amazon Web Services, Azure |
| **Data level** | • Country, city (region) and company levels | • Country and region level |
| **Data references** | • Government reports, carbon footprint website, environmental agency, papers, company reports, etc. | • Carbon footprint website |
| **Update version** | • Updating in 2020 & 2021 | • Updating to 2022 |

**The goal of these tools is** to make people aware of the carbon emission impact, to provide a quick tool to evaluate the carbon emission during machine learning work, and to recommend carbon reduction actions like selecting the cloud provider/location wisely, buying carbon offsets, choosing clean energy, and improving AI algorithms to be green.

# Outline

- Analysis of measurement tools
- Case study: Re-ID
- Conclusions & future work

# AI use case: ReID

## Major AI model types for unstructured data and usages:

### CNN

CV(image/video classification, recognition, search…), NLP(voice recognition, text classification…), …

### LSTM

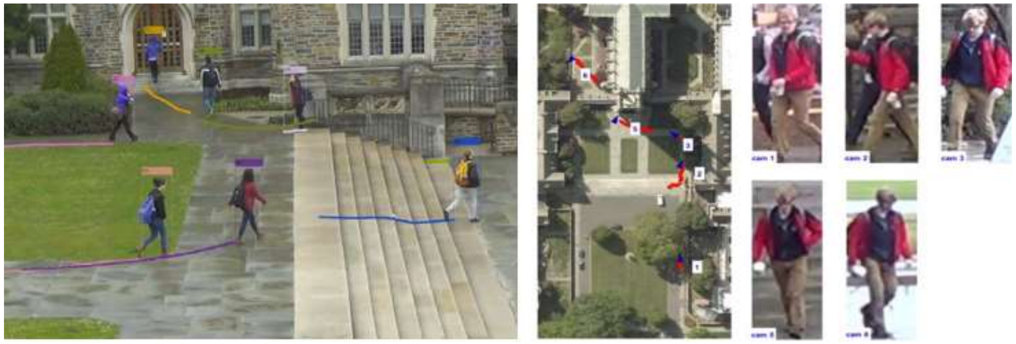
Time series, failure detection, forecasting, NLP(voice recognition, machine translation, text generation…), CV(image classification, image caption generation…), …

### Transformer

CV(image caption generation, target detection, image classification, …), NLP(chatbot, machine translation, text generation…), …

## What is ReID?

Re-Identification is the task of associating images of the same object (person/vehicle) taken from different cameras or from the same camera in different occasions.



# Setting for experiments

### Input testing video for inference:

- Single: only one person in the video
- Multi: there are always more than two person in the video
- Mixed: dynamic picture, sometimes one person, sometimes many people, and sometimes no one

### Scenarios definition:

- Fixed time: running AI application for 500s
- Fixed task: using/analysis a same 5mins video as input, stop application until all frame finished

### KPI for performance:

- FPS: the processing speed of the video
- Accuracy: the accuracy of video processing (identifying people)

### Units:

- Second s for time
- Wh for power consumption

### Server: GPU:

- GPU: GTX 1080 Ti ; RTX 3080 Ti (for training)
- CPU: Intel Xeon E5-2678 v3; Intel i9-12900H (for training)
- Memory: 64GB; 32GB (for training)

### Jetson Xavier:

- GPU: NVDIA Jetson AGX Xavier
- CPU: ARMv8 Processor rev 0 (v8l)
- Memory: 32GB

### Intel NUC:

- GPU: NO
- CPU: Intel i7-8559U
- Memory: 16GB

# Benchmark: experiment vs estimation

## Fast-ReID, CNN, Single person, running time 500s, inference

| Measurement tools | GPU/CPU type | Power consumption | | | | Carbon emissions[a] |
|---|---|---|---|---|---|---|
| | | CPU (W) | Usage | Memory (GB) | Total (Wh) | CO2e(mg) |
| 1 On the fly - PyJoules | Server:<br>GPU: GeForce GTX 1080 Ti<br>CPU: Intel Xeon E5-2678 v3<br>Memory: 64GB | | | | 7.2 | |
| 2 A posteriori - MLCO2 Impact | | 120 | 100% | | 16.67 | 633.46 |
| 3 A posteriori – Green Algorithms | | 120 | 30%/100% | 64 | 8.31/19.98 | 426.18/ 1002 |
| 1 On the fly - PyJoules | Intel machine II:<br>CPU: Intel i7-8559U<br>Memory: 16GB | | | | 4.1 | |
| 2 A posteriori - MLCO2 Impact | | 28 | 100% | | 3.89 | 147.82 |
| 3 A posteriori – Green Algorithms | | 28 | 70%/100% | 16 | 3.55/4.72 | 182.04/ 241.86 |
| 1 On the fly - Jtop | ARM:<br>NVDIA Jetson AGX Xavier<br>CPU: ARMv8 Processor rev 0 (v8l)<br>Memory: 32GB | | | | 3.8 | |
| 2 A posteriori - MLCO2 Impact | | 30 | 100% | | 4.17 | 158.46 |
| 3 A posteriori – Green Algorithms | | 30 | 70%/100% | 32 | 4.57/5.82 | 234.45/ 298.55 |

a. The reference location is Europe, France

# Benchmark: experiment vs estimation

**Server:**
- GPU: GeForce RTX 3080 Ti
- CPU: i9-12900H
- Memory: 32GB

**Measurement tools:**
1. On the fly – PyJoules
2. A posteriori - MLCO2 Impact
3. A posteriori – Green Algorithms

## Training

| Test cases | Measurement tools | Running time(s) | Power consumption | | | | | | Carbon emissions[a] |
|---|---|---|---|---|---|---|---|---|---|
| | | | GPU (W) | Usage | CPU (W) | Usage | Memory (GB) | Total (Wh) | CO2e(g) |
| 1 | Fast-ReID, CNN | 4625 | | | | | | 125.06 | |
| 2 | | | 120 | 100% | 45 | 100% | | 211.98 | 8.06 |
| 3 | | | 120 | 66%/100% | 45 | 10%/100% | 64 | 128.16/242.61 | 7.08/12.44 |
| 1 | st-ReID, CNN | 7988 | | | | | | 212.26 | |
| 2 | | | 120 | 100% | 45 | 100% | | 366.12 | 13.91 |
| 3 | | | 120 | 66%/100% | 45 | 10%/100% | 64 | 238.62/419.01 | 12.24/21.49 |
| 1 | DeepPerson, LSTM | 8326 | | | | | | 201.74 | |
| 2 | | | 120 | 100% | 45 | 100% | | 381.61 | 30.35 |
| 3 | | | 120 | 66%/100% | 45 | 10%/100% | 64 | 248.72/436.74 | 26.69/46.87 |
| 1 | Trans-ReID, Transformer | 17426 | | | | | | 451.7 | |
| 2 | | | 120 | 100% | 45 | 100% | | 798.69 | 30.35 |
| 3 | | | 120 | 66%/100% | 45 | 10%/100% | 64 | 520.55/914.09 | 26.69g/46.87 |

a. The reference location is Europe, France

# Comments in the training phase

**Total training power consumption is determined by the training algorithm and training time. Training power consumption is in proportion to the training time in general.  With the measurement tool, it can quantify power consumption in order to make more accurate assessment for different AI models.**

**The complexity of model:**

- **Trans-ReID(transformer) ≈ st-ReID(CNN) > Deepperson (LSTM) > fast-ReID (CNN)**

**Considering to performance and energy saving:**

- **The training program runs on GPU generally, and in the training phase fast-ReID is the best choice for a balanced requirement of performance and energy saving.**
- **Trans-ReID is the newest model probably without optimized for energy consumption. We hope it can be optimized on energy consumption in the future.**
- **Our recommended order according to training phase**
  - fast-ReID (CNN), st-ReID(CNN), Trans-ReID(transformer), Deepperson (LSTM)

# Outline

- Analysis of measurement tools
- Case study: Re-ID
- Conclusions & future work

# Conclusions & future work

## Conclusions

- **The effectiveness of priori measurement tools relies on their detailed implementation. The application of priori measurement tools is limited. The tools just support one special framework and a subset of types of model layers.**
- **The on-the-fly tools can be used during the processes of AI programs; however, they are limited. the comparison of experimental and estimated results shows that the error of the on-the-fly measurement tools is acceptable.**
- **The posteriori measurement tools can be used for power consumption estimation after the AI processing by knowing the runtime and the parameters of hardware (CPU, GPU, memory, etc.).**
- **The experimental results show that the total training power consumption of the AI model is determined by the training algorithm and training time. Training power consumption is in proportion to the training time in general.  With the measurement tool, it can quantify the power consumption to make a more accurate assessment for different AI models.**

## Future work:

- **continuously investigating and improving the measurement tools and verifying them in experiments.**
- **with these tools, researchers and scientists may be able to design more power-efficient AI models without sacrificing model performance.**

# Thanks