Lecture 1 - Energy Consumption in Machine Learning

Efficacy and efficiency evaluation of machine learning models
Ph.D. Course

Marco Frasca

AnacletoLab, Dipartimento di Informatica
Università degli Studi di Milano

11.06.24

Outline

1. The Increase of Power Demand in ML

2. Estimating the Energy Consumption of ML Algorithms
   - 2.1 Empirical tools to estimate the effective hardware usage
   - 2.2 Theoretical Models of Energy Consumption
   - 2.3 ML estimators of energy consumption

Outline

1. The Increase of Power Demand in ML

2. Estimating the Energy Consumption of ML Algorithms
   - 2.1 Empirical tools to estimate the effective hardware usage
   - 2.2 Theoretical Models of Energy Consumption
   - 2.3 ML estimators of energy consumption

The Increase of Power Demand in Machine Learning (ML)

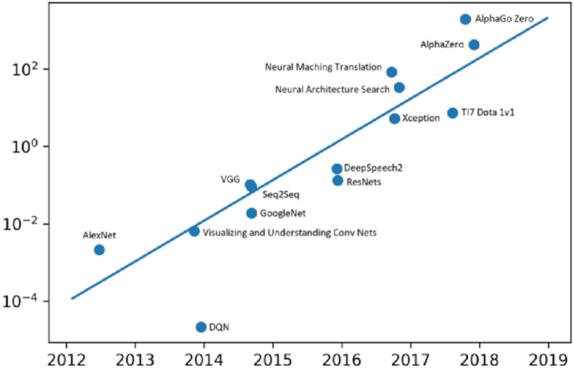
- The increasing size of ML models is a problem

The Increase of Power Demand in Machine Learning (ML)

- The increasing size of ML models is a problem
- Severely increased computational complexity

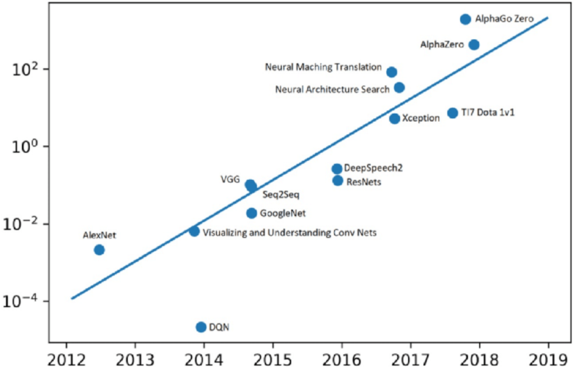
The Increase of Power Demand in Machine Learning (ML)

- The increasing size of ML models is a problem
- Severely increased computational complexity
- Need for power-hungry hardware (petaflops/day, in 1e04)



The Increase of Power Demand in Machine Learning (ML)

- The increasing size of ML models is a problem
- Severely increased computational complexity
- Need for power-hungry hardware (petaflops/day, in 1e04)



Amodei, D. and Hernandez, D. Ai and compute. https://openai.com/index/ai-and-compute/ 2018.

- To train *AlphaGO Zero* it took 4 TPUs, 64 GPUs and 19 CPUs for days!! Around 3 millions of $!!!

Growth in Required Compute

Bruckner, T., et *al*. Energy systems. In Climate Change 2014: Mitigation of Climate Change.

- In 2010 energy production was responsible for approximately 35% of total anthropogenic greenhouse gas (GHG) emissions

Growth in Required Compute

Bruckner, T., et *al*. Energy systems. In Climate Change 2014: Mitigation of Climate Change.

- In 2010 energy production was responsible for approximately 35% of total anthropogenic greenhouse gas (GHG) emissions
- If this exponential trend continue, ML and mainly Deep Learning (DL) compute may become a significant contributor to climate change

Growth in Required Compute

Bruckner, T., et *al*. Energy systems. In Climate Change 2014: Mitigation of Climate Change.

- In 2010 energy production was responsible for approximately 35% of total anthropogenic greenhouse gas (GHG) emissions
- If this exponential trend continue, ML and mainly Deep Learning (DL) compute may become a significant contributor to climate change
- This can be mitigated by exploring how to improve energy efficiency in ML

Need to Change of the Assessment Paradigm

Strubell, E., Ganesh, A., and McCallum, A. Energy and Policy Considerations for Deep Learning in NLP. pp. 3645–3650, 2019. doi: 10.18653/v1/p19-1355.

- The trade-off performance/environmental-impact should still be taken into account

Need to Change of the Assessment Paradigm

Strubell, E., Ganesh, A., and McCallum, A. Energy and Policy Considerations for Deep Learning in NLP. pp. 3645–3650, 2019. doi: 10.18653/v1/p19-1355.

- The trade-off performance/environmental-impact should still be taken into account
- Ongoing trends recommend that metrics such as training and inference time, computational resources required, and model sensitivity to hyperparameters should be reported to enable direct comparison between models [Strubell et *al.* 2019].

Need to Change of the Assessment Paradigm

Strubell, E., Ganesh, A., and McCallum, A. Energy and Policy Considerations for Deep Learning in NLP. pp. 3645–3650, 2019. doi: 10.18653/v1/p19-1355.

- The trade-off performance/environmental-impact should still be taken into account
- Ongoing trends recommend that metrics such as training and inference time, computational resources required, and model sensitivity to hyperparameters should be reported to enable direct comparison between models [Strubell et *al.* 2019].
- Hot subject for scientific meetings and conferences, which ask to include results about ML energy consumption estimate
  - E.g. Workshop on Simplification, Compression, Efficiency and Frugality for Artificial intelligence (ECML PKDD 2023)

Need to Change of the Assessment Paradigm

Strubell, E., Ganesh, A., and McCallum, A. Energy and Policy Considerations for Deep Learning in NLP. pp. 3645–3650, 2019. doi: 10.18653/v1/p19-1355.

- The trade-off performance/environmental-impact should still be taken into account
- Ongoing trends recommend that metrics such as training and inference time, computational resources required, and model sensitivity to hyperparameters should be reported to enable direct comparison between models [Strubell et *al.* 2019].
- Hot subject for scientific meetings and conferences, which ask to include results about ML energy consumption estimate
  - E.g. Workshop on Simplification, Compression, Efficiency and Frugality for Artificial intelligence (ECML PKDD 2023)
- **PROBLEM**: lack of tools to appropriately compute the energy consumption
  - Accounting for all involved factor is almost impossible, and several assumptions are needed

Outline

1. The Increase of Power Demand in ML

2. Estimating the Energy Consumption of ML Algorithms
   - 2.1 Empirical tools to estimate the effective hardware usage
   - 2.2 Theoretical Models of Energy Consumption
   - 2.3 ML estimators of energy consumption

Estimating the Energy Consumption of ML Algorithms

Three main approaches:

1. Empirical tools to estimate the effective hardware usage (e.g.,CarbonTracker, Codecarbon)
   - Energy consumption heavily depends on the energy efficiency of the hardware

Estimating the Energy Consumption of ML Algorithms

Three main approaches:

1. Empirical tools to estimate the effective hardware usage (e.g.,CarbonTracker, Codecarbon)
   - Energy consumption heavily depends on the energy efficiency of the hardware
2. Theoretical models of energy consumption (based on theoretical algorithm analyses)
   - Consumption estimated regardless the used hardware
   - Modelling based on abstracting elementary operations

Estimating the Energy Consumption of ML Algorithms

Three main approaches:

1. Empirical tools to estimate the effective hardware usage (e.g.,CarbonTracker, Codecarbon)
   - Energy consumption heavily depends on the energy efficiency of the hardware
2. Theoretical models of energy consumption (based on theoretical algorithm analyses)
   - Consumption estimated regardless the used hardware
   - Modelling based on abstracting elementary operations
3. ML estimators of energy ML algorithms consumption (what?!?)

For a survey:
García-Martín, E. et *al*. Estimation of energy consumption in machine learning, Journal of Parallel and Distributed Computing, 134, 2019,pp 75-88.

Outline

1. The Increase of Power Demand in ML

2. Estimating the Energy Consumption of ML Algorithms
   - 2.1 Empirical tools to estimate the effective hardware usage
   - 2.2 Theoretical Models of Energy Consumption
   - 2.3 ML estimators of energy consumption

Empirical tools to estimate the effective hardware usage

- Some tools to estimate code carbon footprint:
  - *Machine Learning Emissions Calculator*. The tool can estimate the carbon footprint of GPU compute by specifying hardware type, hours used, cloud provider, and region [Lacoste 2019].
    - Available as web tool (`https://mlco2.github.io/impact/`)
  - *Experiment-impact-tracker* [Henderson 2020]
  - *CarbonTracker* [Anthony et *al.*]
  - *Tracarbon* [`https://github.com/fvaleye/tracarbon`]
  - ....

Empirical tools to estimate the effective hardware usage

- Some tools to estimate code carbon footprint:
  - *Machine Learning Emissions Calculator*. The tool can estimate the carbon footprint of GPU compute by specifying hardware type, hours used, cloud provider, and region [Lacoste 2019].
    - Available as web tool (`https://mlco2.github.io/impact/`)
  - *Experiment-impact-tracker* [Henderson 2020]
  - *CarbonTracker* [Anthony et *al.*]
  - *Tracarbon* [`https://github.com/fvaleye/tracarbon`]
  - ....
- They need to know the cloud resource provider (Google Cloud Provider, Amazon Web Services, etc.) and the country
  - Different providers and different countries produce electricity by using different sources, with different emissions

– Lacoste, A.et *al*. Quantifying the Carbon Emissions of Machine – Learning. Technical report, 2019.

– Henderson, P., et *al.* Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning. http://arxiv.org/abs/2002.05651. 2020

– Anthony LFW. et *al*. Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models, ICML Workshop on Challenges in Deploying and monitoring Machine Learning Systems, 2020

CodeCarbon

We will try to go a bit deeper about one of such tools, CodeCarbon [`https://codecarbon.io`]

- Python pip package

CodeCarbon

We will try to go a bit deeper about one of such tools, CodeCarbon [https://codecarbon.io]

- Python pip package
- Easy to integrate in your Python code

CodeCarbon

We will try to go a bit deeper about one of such tools, CodeCarbon [https://codecarbon.io]

- Python pip package
- Easy to integrate in your Python code
- It estimates the amount of carbon dioxide (CO_2) produced by the cloud or personal computing resources used to execute the code

CodeCarbon

We will try to go a bit deeper about one of such tools, CodeCarbon [https://codecarbon.io]

- Python pip package
- Easy to integrate in your Python code
- It estimates the amount of carbon dioxide (CO_2) produced by the cloud or personal computing resources used to execute the code
- Effective visualization of outputs in an integrated dashboard

CodeCarbon: Carbon Dioxide Emissions Estimation

Carbon dioxide (CO_2) emissions are estimated as the product of two main factors:

1. Carbon Intensity of the electricity consumed for computation, quantified as g of CO_2 emitted per kilowatt-hour

CodeCarbon: Carbon Dioxide Emissions Estimation

Carbon dioxide (CO_2) emissions are estimated as the product of two main factors:

1. Carbon Intensity of the electricity consumed for computation, quantified as g of CO_2 emitted per kilowatt-hour
   - Calculated as a weighted average of the emissions from the different energy sources
   - Each source has its own carbon intensity

CodeCarbon: Carbon Dioxide Emissions Estimation

Carbon dioxide (CO_2) emissions are estimated as the product of two main factors:

1. Carbon Intensity of the electricity consumed for computation, quantified as g of CO_2 emitted per kilowatt-hour
   - Calculated as a weighted average of the emissions from the different energy sources
   - Each source has its own carbon intensity
2. Energy Consumed by the computational infrastructure, quantified as kilowatt-hours

CodeCarbon: Carbon Intensity (CI)

- Like most tools of this type, CodeCarbon computes the carbon intensity of electricity per cloud provider and per country
  - based on public data of each country about the energy source composition (and their relative CO_2 emissions): e.g. *biofuel*, *coal*, *fossil*, *gas*, *hydroelectricity*, *nuclear*, *solar*, *wind*

CodeCarbon: Carbon Intensity (CI)

- Like most tools of this type, CodeCarbon computes the carbon intensity of electricity per cloud provider and per country
  - based on public data of each country about the energy source composition (and their relative CO_2 emissions): e.g. *biofuel*, *coal*, *fossil*, *gas*, *hydroelectricity*, *nuclear*, *solar*, *wind*
- Average CI per kWh varies with the composition of sources

CodeCarbon: Carbon Intensity (CI)

- Like most tools of this type, CodeCarbon computes the carbon intensity of electricity per cloud provider and per country
  - based on public data of each country about the energy source composition (and their relative CO_2 emissions): e.g. *biofuel*, *coal*, *fossil*, *gas*, *hydroelectricity*, *nuclear*, *solar*, *wind*
- Average CI per kWh varies with the composition of sources
- The mix varies by country (there are tables available)

CodeCarbon: Carbon Intensity (CI)

- Like most tools of this type, CodeCarbon computes the carbon intensity of electricity per cloud provider and per country
  - based on public data of each country about the energy source composition (and their relative CO_2 emissions): e.g. *biofuel*, *coal*, *fossil*, *gas*, *hydroelectricity*, *nuclear*, *solar*, *wind*
- Average CI per kWh varies with the composition of sources
- The mix varies by country (there are tables available)
- When carbon intensity is not available, but the energy mix yes, it is assumed to be following average:

Energy Source	**Carbon Intensity (kg/kWh)**
Coal	995
Petroleum	816
Natural Gas	743
Geothermal	38
Hydroelectricity	26
Nuclear	29
Solar	48
Wind	26

Source: Codecarbon https://mlco2.github.io/codecarbon/methodology.html

CodeCarbon: Carbon Intensity

Energy Source	Carbon Intensity (g/kWh)
Coal	995
Petroleum	816
Natural Gas	743
Geothermal	38
Hydroelectricity	26
Nuclear	29
Solar	48
Wind	26

Then, for example, if the energy mix of a grid electricity is 25% Coal, 35% Petroleum, 26% Natural Gas and 14% Nuclear, but no carbon intensity in known for that region, we get
Net Carbon Intensity =

$$0.25 \cdot 995 + 0.35 \cdot 816 + 0.26 \cdot 743 + 0.14 \cdot 29 = 731.59\ gCO_2/kWh$$

CodeCarbon: Carbon Intensity

Energy Source	Carbon Intensity (g/kWh)
Coal	995
Petroleum	816
Natural Gas	743
Geothermal	38
Hydroelectricity	26
Nuclear	29
Solar	48
Wind	26

Then, for example, if the energy mix of a grid electricity is 25% Coal, 35% Petroleum, 26% Natural Gas and 14% Nuclear, but no carbon intensity in known for that region, we get
Net Carbon Intensity =

$$0.25 \cdot 995 + 0.35 \cdot 816 + 0.26 \cdot 743 + 0.14 \cdot 29 = 731.59\ gCO_2/kWh$$

- If neither the global carbon intensity of a country nor its electricity mix is available, they apply a world average carbon intensity per kilowatt/hour of 475 gCO_2

Power Usage

- It is another difficult task
  - depends on several factors: operative system type, the type of hardware, the presence of multiprocessing, etc.

Power Usage

- It is another difficult task
  - depends on several factors: operative system type, the type of hardware, the presence of multiprocessing, etc.
- The underlying hardware is tracked at frequent time intervals
  - This is a configurable parameter `measure_power_secs`

Power Usage

- It is another difficult task
  - depends on several factors: operative system type, the type of hardware, the presence of multiprocessing, etc.
- The underlying hardware is tracked at frequent time intervals
  - This is a configurable parameter `measure_power_secs`
- Currently, the package supports the following hardware:

Power Usage

- It is another difficult task
  - depends on several factors: operative system type, the type of hardware, the presence of multiprocessing, etc.
- The underlying hardware is tracked at frequent time intervals
  - This is a configurable parameter `measure_power_secs`
- Currently, the package supports the following hardware:

  1. *GPU*, tracks Nvidia GPUs energy consumption using *pynvml* library

Power Usage

- It is another difficult task
  - depends on several factors: operative system type, the type of hardware, the presence of multiprocessing, etc.
- The underlying hardware is tracked at frequent time intervals
  - This is a configurable parameter `measure_power_secs`
- Currently, the package supports the following hardware:

1. *GPU*, tracks Nvidia GPUs energy consumption using *pynvml* library
2. *RAM*, codecarbon uses 3 Watts for 8 GB ratio as a gross estimate

Power Usage

- It is another difficult task
  - depends on several factors: operative system type, the type of hardware, the presence of multiprocessing, etc.
- The underlying hardware is tracked at frequent time intervals
  - This is a configurable parameter `measure_power_secs`
- Currently, the package supports the following hardware:
  1. *GPU*, tracks Nvidia GPUs energy consumption using *pynvml* library
  2. *RAM*, codecarbon uses 3 Watts for 8 GB ratio as a gross estimate
  3. *CPU*, here the processors energy consumption is differentiated between Intel, Apple Silicon Chips (M1, M2) and AMD processor (see docs for details)
- Let's move to the notebook

References: https://arxiv.org/pdf/1911.08354

Outline

1. The Increase of Power Demand in ML

2. Estimating the Energy Consumption of ML Algorithms
   - 2.1 Empirical tools to estimate the effective hardware usage
   - 2.2 Theoretical Models of Energy Consumption
   - 2.3 ML estimators of energy consumption

Theoretical Models of Energy Consumption

- Modelling the energy consumption of the base of elementary operations
- Such an energy model is hardware-independent

Theoretical Models of Energy Consumption

- Modelling the energy consumption of the base of elementary operations
- Such an energy model is hardware-independent
  - It can be adapted to specific hardware platforms
  - Need to estimate the cost of individual operations

Theoretical Models of Energy Consumption

- Modelling the energy consumption of the base of elementary operations
- Such an energy model is hardware-independent
  - It can be adapted to specific hardware platforms
  - Need to estimate the cost of individual operations
- The gross estimate is related to the cost of layer computation from inputs to outputs

Theoretical Models of Energy Consumption

- Modelling the energy consumption of the base of elementary operations
- Such an energy model is hardware-independent
  - It can be adapted to specific hardware platforms
  - Need to estimate the cost of individual operations
- The gross estimate is related to the cost of layer computation from inputs to outputs
- Need to analyze the pseudocode of layer computations to estimate the associated energy consumption

Theoretical Models of Energy Consumption

- Modelling the energy consumption of the base of elementary operations
- Such an energy model is hardware-independent
  - It can be adapted to specific hardware platforms
  - Need to estimate the cost of individual operations
- The gross estimate is related to the cost of layer computation from inputs to outputs
- Need to analyze the pseudocode of layer computations to estimate the associated energy consumption
- Then, the energy of one run is the sum of the energy costs of all layers involved from input to output

Theoretical Models of Energy Consumption

- Modelling the energy consumption of the base of elementary operations
- Such an energy model is hardware-independent
  - It can be adapted to specific hardware platforms
  - Need to estimate the cost of individual operations
- The gross estimate is related to the cost of layer computation from inputs to outputs
- Need to analyze the pseudocode of layer computations to estimate the associated energy consumption
- Then, the energy of one run is the sum of the energy costs of all layers involved from input to output
- More suitable to estimate the energy cost of one inference than than of training (it needs to be adapted)

See for instance:

– Brooks, D. et *al.*. Wattch: a framework for architectural-level power analysis and optimizations. SIGARCH Comput. Archit. 28:2 2000, 83–94.

– W. Ye,et *al.*. The design and use of simplepower: a cycle-accurate energy estimation tool. DAC '00, pp. 340–345.

Theoretical Models of Energy Consumption: An Example

Wiedemann, S. Müller, K.R. and Samek, W. Compact and computationally efficient representation of deep neural networks. IEEE Transactions on Neural Networks and Learning Systems, 31(3):772–785, 2020.

- Four elementary operations:
  mul, the binary multiplication operator,
  sum, the binary addition operator,
  read, which reads a value from memory, and
  write, which writes a value into memory.

Theoretical Models of Energy Consumption: An Example

Wiedemann, S. Müller, K.R. and Samek, W. Compact and computationally efficient representation of deep neural networks. IEEE Transactions on Neural Networks and Learning Systems, 31(3):772–785, 2020.

- Four elementary operations:
  mul, the binary multiplication operator,
  sum, the binary addition operator,
  read, which reads a value from memory, and
  write, which writes a value into memory.

- This model assimilates directly into the costs of the corresponding elementary operations the costs of read/write operations from/into low-level memory (like caches and registers) that stores temporary runtime values

Theoretical Models of Energy Consumption: An Example

Wiedemann, S. Müller, K.R. and Samek, W. Compact and computationally efficient representation of deep neural networks. IEEE Transactions on Neural Networks and Learning Systems, 31(3):772–785, 2020.

- Four elementary operations:
  mul, the binary multiplication operator,
  sum, the binary addition operator,
  read, which reads a value from memory, and
  write, which writes a value into memory.

- This model assimilates directly into the costs of the corresponding elementary operations the costs of read/write operations from/into low-level memory (like caches and registers) that stores temporary runtime values

- The energy requirement for a layer computation is expressed in terms of such 4 operations

Theoretical Models of Energy Consumption: An Example

Wiedemann, S. Müller, K.R. and Samek, W. Compact and computationally efficient representation of deep neural networks. IEEE Transactions on Neural Networks and Learning Systems, 31(3):772–785, 2020.

- Four elementary operations:
  mul, the binary multiplication operator,
  sum, the binary addition operator,
  read, which reads a value from memory, and
  write, which writes a value into memory.

- This model assimilates directly into the costs of the corresponding elementary operations the costs of read/write operations from/into low-level memory (like caches and registers) that stores temporary runtime values

- The energy requirement for a layer computation is expressed in terms of such 4 operations

- Then, each elementary operation is associated with an energy cost, estimated from hardware

Theoretical Models of Energy Consumption: An Example

Horowitz, M. 1.1 Computing's Energy Problem (and what we can do about it). In ISSCC, pp 10–14, 2014.

In the paper above the y provide the following estimate of elementary operations energy cost for a 45-nm CMOS processor:

Table: Energy is in pJ (Picojoule). MB and KB denotes megabytes and kilobytes, respectively.

Operation	**8 bits**	**16 bits**	**32 bits**
float add	0.2	0.4	0.9
float mul	0.6	1.1	3.7
R/W (<8KB)	1.25	2.5	5.0
R/W (<32KB)	2.5	5.0	10.0
R/W (<1MB)	12.5	25.0	50.0
R/W (>1MB)	250.0	500.0	1000.0

- Pros: Changing the hardware needs just to calibrate the table

Theoretical Models of Energy Consumption: An Example

Horowitz, M. 1.1 Computing's Energy Problem (and what we can do about it). In ISSCC, pp 10–14, 2014.

In the paper above the y provide the following estimate of elementary operations energy cost for a 45-nm CMOS processor:

Table: Energy is in pJ (Picojoule). MB and KB denotes megabytes and kilobytes, respectively.

Operation	**8 bits**	**16 bits**	**32 bits**
float add	0.2	0.4	0.9
float mul	0.6	1.1	3.7
R/W (<8KB)	1.25	2.5	5.0
R/W (<32KB)	2.5	5.0	10.0
R/W (<1MB)	12.5	25.0	50.0
R/W (>1MB)	250.0	500.0	1000.0

- Pros: Changing the hardware needs just to calibrate the table
- Cons: Reducing the computation to just four operations discards other costs that be relevant

Theoretical Models of Energy Consumption: An Example

Horowitz, M. 1.1 Computing's Energy Problem (and what we can do about it). In ISSCC, pp 10–14, 2014.

In the paper above the y provide the following estimate of elementary operations energy cost for a 45-nm CMOS processor:

Table: Energy is in pJ (Picojoule). MB and KB denotes megabytes and kilobytes, respectively.

Operation	**8 bits**	**16 bits**	**32 bits**
float add	0.2	0.4	0.9
float mul	0.6	1.1	3.7
R/W (<8KB)	1.25	2.5	5.0
R/W (<32KB)	2.5	5.0	10.0
R/W (<1MB)	12.5	25.0	50.0
R/W (>1MB)	250.0	500.0	1000.0

- Pros: Changing the hardware needs just to calibrate the table
- Cons: Reducing the computation to just four operations discards other costs that be relevant
- But to compare algorithms for the same task that's not a problem

Outline

1. The Increase of Power Demand in ML

2. Estimating the Energy Consumption of ML Algorithms
   - 2.1 Empirical tools to estimate the effective hardware usage
   - 2.2 Theoretical Models of Energy Consumption
   - 2.3 ML estimators of energy consumption

ML to estimate Energy Consumption of ML Algorithm

- (J. Getzner et *al*. ICLR 2023) Accuracy is not the only Metric that Matters: Estimating the Energy Consumption of Deep Learning Models
  - Pipeline to estimate the energy consumption of a model

ML to estimate Energy Consumption of ML Algorithm

- (J. Getzner et *al*. ICLR 2023) Accuracy is not the only Metric that Matters: Estimating the Energy Consumption of Deep Learning Models
  - Pipeline to estimate the energy consumption of a model
  - Before training and without the need for a direct energy measurement

ML to estimate Energy Consumption of ML Algorithm

- (J. Getzner et *al*. ICLR 2023) Accuracy is not the only Metric that Matters: Estimating the Energy Consumption of Deep Learning Models
  - Pipeline to estimate the energy consumption of a model
  - Before training and without the need for a direct energy measurement
  - A data set of energy consumption of various models and layer types

ML to estimate Energy Consumption of ML Algorithm

- (J. Getzner et *al.* ICLR 2023) Accuracy is not the only Metric that Matters: Estimating the Energy Consumption of Deep Learning Models
  - Pipeline to estimate the energy consumption of a model
  - Before training and without the need for a direct energy measurement
  - A data set of energy consumption of various models and layer types
  - A collection of predictors for different layer types, forming an energy-prediction baseline to more complex Deep Learning architectures

Getzner, J. Charpentier, B. and Günnemann, S.. (2023). Accuracy is not the only Metric that matters: Estimating the Energy Consumption of Deep Learning Models. arXiv:2304.00897.

Accuracy is not the only Metric that matters: Estimating the Energy Consumption of Deep Learning Models

- Different parameters considered for each layer (when present): *batch-size*, *image-size*, *kernel-size*, *in-channels/size*, *out-channels/size*, *stride*, *padding*

Accuracy is not the only Metric that matters: Estimating the Energy Consumption of Deep Learning Models

- Different parameters considered for each layer (when present): *batch-size*, *image-size*, *kernel-size*, *in-channels/size*, *out-channels/size*, *stride*, *padding*
- Different layer types: *Conv2d, MaxPool2D, Linear*

Accuracy is not the only Metric that matters: Estimating the Energy Consumption of Deep Learning Models

- Different parameters considered for each layer (when present): *batch-size*, *image-size*, *kernel-size*, *in-channels/size*, *out-channels/size*, *stride*, *padding*
- Different layer types: *Conv2d, MaxPool2D, Linear*
- Different activation functions: *ReLU, Sigmoid, Tanh, Softmax*

Accuracy is not the only Metric that matters: Estimating the Energy Consumption of Deep Learning Models

- Different parameters considered for each layer (when present): *batch-size*, *image-size*, *kernel-size*, *in-channels/size*, *out-channels/size*, *stride*, *padding*
- Different layer types: *Conv2d, MaxPool2D, Linear*
- Different activation functions: *ReLU, Sigmoid, Tanh, Softmax*
- Different architectures: AlexNet (Krizhevsky et *al*., 2017), and VGG11/13/16

Accuracy is not the only Metric that matters: Estimating the Energy Consumption of Deep Learning Models

- Different parameters considered for each layer (when present): *batch-size*, *image-size*, *kernel-size*, *in-channels/size*, *out-channels/size*, *stride*, *padding*
- Different layer types: *Conv2d, MaxPool2D, Linear*
- Different activation functions: *ReLU, Sigmoid, Tanh, Softmax*
- Different architectures: AlexNet (Krizhevsky et *al.*, 2017), and VGG11/13/16
- All required parameters are sampled randomly from configurable ranges

Accuracy is not the only Metric that matters: Estimating the Energy Consumption of Deep Learning Models

- Different parameters considered for each layer (when present): *batch-size*, *image-size*, *kernel-size*, *in-channels/size*, *out-channels/size*, *stride*, *padding*
- Different layer types: *Conv2d, MaxPool2D, Linear*
- Different activation functions: *ReLU, Sigmoid, Tanh, Softmax*
- Different architectures: AlexNet (Krizhevsky et *al.*, 2017), and VGG11/13/16
- All required parameters are sampled randomly from configurable ranges
- Use Codecarbon to measure the CPU energy of the randomly configured models obtained

– Krizhevsky, A. Sutskever, I. and Hinton, G.E. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6):84–90, 2017.

Accuracy is not the only Metric that matters: Estimating the Energy Consumption of Deep Learning Models

- They use Linear/Polynomial Regression models to predict the CPU energy consumption of each layer Different layer types: *Conv2d, MaxPool2D, Linear*

Accuracy is not the only Metric that matters: Estimating the Energy Consumption of Deep Learning Models

- They use Linear/Polynomial Regression models to predict the CPU energy consumption of each layer Different layer types: *Conv2d, MaxPool2D, Linear*
  - They did not expect any higher-order dependencies or strong non-linear relationships omong features
  - Used models have superior transparency and interpretablity

Accuracy is not the only Metric that matters: Estimating the Energy Consumption of Deep Learning Models

- They use Linear/Polynomial Regression models to predict the CPU energy consumption of each layer Different layer types: *Conv2d, MaxPool2D, Linear*
  - They did not expect any higher-order dependencies or strong non-linear relationships omong features
  - Used models have superior transparency and interpretablity
- For linear (FC) and convolutional layers, the multiply accumulate count (MAC) as the only feature and achieved an R^2 test score > 0.999!!

Accuracy is not the only Metric that matters: Estimating the Energy Consumption of Deep Learning Models

- Convolutional layers contributed to $> 80\%$ of the total energy consumption, beacause most layers were convolutional

Accuracy is not the only Metric that matters: Estimating the Energy Consumption of Deep Learning Models

- Convolutional layers contributed to $> 80\%$ of the total energy consumption, beacause most layers were convolutional
- But FC in general layers tend to have much more parameters than convolutional ones
- A few of them would largely impact on the energy consumption

Accuracy is not the only Metric that matters: Estimating the Energy Consumption of Deep Learning Models

- Convolutional layers contributed to $> 80\%$ of the total energy consumption, beacause most layers were convolutional
- But FC in general layers tend to have much more parameters than convolutional ones
- A few of them would largely impact on the energy consumption
- Main limitation: only CPU estimates are available