Quantifying the Carbon Emissions of Machine Learning

Alexandre Lacoste*
Element AI
allac@elementai.com

Alexandra Luccioni*
Mila, Université de Montréal
luccionis@mila.quebec

Victor Schmidt*
Mila, Université de Montréal
schmidtv@mila.quebec

Thomas Dandres
Polytechnique Montréal, CIRAIG
thomas.dandres@polymtl.ca

Abstract

From an environmental standpoint, there are a few crucial aspects of training a neural network that have a major impact on the quantity of carbon that it emits. These factors include: the location of the server used for training and the energy grid that it uses, the length of the training procedure, and even the make and model of hardware on which the training takes place. In order to approximate these emissions, we present our *Machine Learning Emissions Calculator*, a tool for our community to better understand the environmental impact of training ML models. We accompany this tool with an explanation of the factors cited above, as well as concrete actions that individual practitioners and organizations can take to mitigate their carbon emissions.

1 Introduction

While a decade ago, only a few ML pioneers were training neural networks on GPUs (Graphical Processing Units), in recent years powerful GPUs have become increasingly accessible and used by ML practitioners worldwide. Furthermore, new models often need to beat existing challenges, which entails training on more GPUs, with larger datasets, for a longer time. This expansion brings with it ever-growing costs in terms of the energy needed to fuel it. This trend has been the subject of recent studies aiming to evaluate the climate impact of AI, which have predominantly put the focus on the environmental cost of training large-scale models connected to grids powered by fossil fuels [1, 2]. While these models are not necessarily representative of common practice, we believe that it is important to continue this conversation further and work towards defining the tools and steps that we need to assess the carbon emissions generated by the models we train, as well as to propose ways to reduce those emissions.

In this work, we present our Machine Learning Emissions Calculator (`https://mlco2.github.io/impact/`), a tool for our community to estimate the amount of carbon emissions produced by training ML models. We accompany this tool with a presentation of key concepts and an explanation of the factors impacting emissions. Finally, we end our article with some recommendations of best practices for the overall ML research community, as well as for individual researchers.

*equal contribution

Preprint. Under review.

2 Quantifying Carbon Emissions in Neural Network Training

In order to quantify carbon emissions, we use *CO_2-equivalents* (CO_2eq), which is a standardized measure used to express the global-warming potential of various greenhouse gases as a single number, i.e. as the amount of CO_2 which would have the equivalent global warming impact [3]. We will use this single metric to compare the factors and choices that impact overall amount of emissions produced by training an ML model in the sections below.

2.1 Type of Energy Used

Practically speaking, it is hard to estimate exactly the amount of CO_2eq emitted by a cloud server in a given location because the information regarding the energy grid that it is connected to is rarely publicly available. However, if we assume that all servers are connected to local grids at their physical location, we are able to make an estimation of the amount of CO_2eq that they emit using public data sources [4, 5] . Therefore, in order to create our emissions calculator, we gathered data regarding CO_2eq emissions of different grid locations and cross-referenced them with known GPU server locations from the three major cloud providers: Google Cloud Platform, Microsoft Azure and Amazon Web Services [2]. Our aim in doing this is to illustrate the degree of variability that exists depending on the location of a given server. For instance, in Figure 1, we show the distribution and variation in carbon emissions depending on geographical region. It can be noted that a large amount of variation can be found within a single region; for instance, servers located in North America can emit anywhere between 20g CO_2eq/kWh in Quebec, Canada to 736.6g CO_2eq/kWh in Iowa, USA [5].

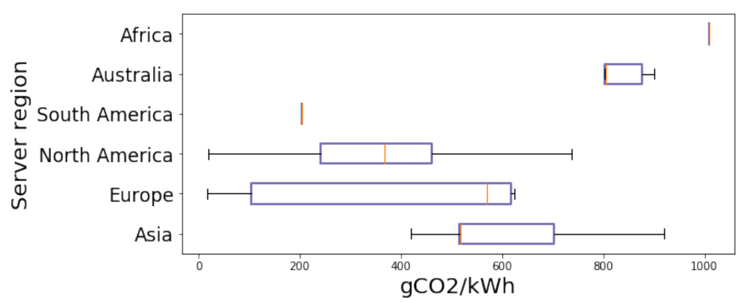

Figure 1: Variation of the Average Carbon Intensity of Servers Worldwide, by Region. (Vertical bars represent regions with a single available data point.)

2.2 Computing Infrastructure and Training Time

Another, more subtle, factor in carbon emitted by a neural network is the computing infrastructure used and training time of the model. In terms of performance the number of floating point operations per second (FLOPS) of GPUs has been steadily increasing in recent years, from 100 Giga FLOPS per second in 2004 to up to 15 Tera FLOPS per second in recent hardware [6]. However, with neural network architectures becoming deeper and more complex, recent state-of-the-art models are often trained on multiples GPUs for several weeks (or months) to beat benchmark performance, requiring more and more energy.

Finally, when it comes to defining a training procedure for ML architectures, there are several elements to consider: for starters, whether it is necessary to train a model from scratch or whether fine-tuning is adequate for the task at hand. Notably, recent research has shown that using pre-trained models with task-specific fine-tuning performs as well as training from scratch, while being more robust, for tasks in image recognition [7, 8] and NLP [9]. Furthermore, when it comes to hyperparameter search, it has been proven both empirically and theoretically that random hyperparameter search is more efficient than grid search for hyperparameter optimization [10], and there is much research being done on ways to improve the efficiency of hyperparameter optimization [11, 12], which makes it

[2]The data can be found at `https://github.com/mlco2/impact/tree/master/data`

possible to continue choosing the right hyperparameters for new models without incurring superfluous computing and energy costs.

3 ML Emissions Calculator and Actionable Items

It is difficult to provide clear-cut guidelines for ML researchers to follow in order to reduce the carbon emissions, or specific benchmarks for the training time that a given model or task warrants. Nonetheless, we think that there are certain best practices and actionable items that can be adopted by our community to reduce environmental impact of the ML domain. We present some of these, along with our ML emissions calculator, in the current section.

Quantify Your Emissions Being informed regarding the factors that impact the quantity of carbon emissions produced by ML research is the first step to making positive changes. It is for this reason that we created our *ML Emissions Calculator*. This tool, currently in its alpha version, takes as input the details regarding the training of an ML model: the geographical zone of the server, the type of GPU, and the training time, and gives as output the approximate amount of CO_2eq produced. We collected publicly available data for the 4 main variables of this computation: (i) the energy consumption of hardware (see "Choose More Efficient Hardware" below), (ii) the location of providers' regions of compute – which we assumed to be connected to their local grid, (iii) the region's CO_2eq emissions per kWh and (iv) potential offsets bought by the provider.

We intend to adopt an open and transparent approach: the data we used is publicly available, debatable and editable through Github issues and pull requests. We are therefore open to updating data as more information becomes available. Since this paper's core goal is to raise awareness around the carbon emissions of ML, we have also included two educational sections in the website: one about learning the main notions and concepts related to this domain (e.g. RECs, carbon neutrality, etc.), the other about actionable items an individual or an organization can leverage to mitigate their carbon impact.

Choose Cloud Providers Wisely In recent years, many cloud providers have defined ambitious sustainability goals and are offsetting their emissions through Renewable Energy Certificates (RECs) in an effort to become carbon neutral, a term used to indicate a net zero carbon footprint of an organization. Each REC bought attests that 1 MWh of renewable energy has been added to the energy grid and can be used to offset an equivalent amount of non-renewable energy. For instance, Google Cloud Platform is certified carbon neutral and funds solar and wind farms directly on local grids through RECs [13]. Microsoft Azure is also certified carbon neutral and 44% of its electricity consumption directly comes from renewable energy, according to a 2016 estimate [14]. Finally, to the best of our knowledge, while not yet 100% carbon neutral on an organizational level, Amazon Web Services is also funding renewable energy projects and some of their data centers are powered by renewable energy [15].

Another major energy consumption factor of server installations is the power usage effectiveness (PUE) of the centers where the GPUs are hosted, which represents the percentage of energy consumption that is used for cooling, power conversion, and other auxiliary tasks, and can vary immensely. For example, Google Cloud Services has an average PUE of 1.1, meaning that only 11% of their total energy usage is not used for the servers themselves, a ratio that they have been steadily reducing using Reinforcement Learning [16, 17]. Finally, if you rely on a local private compute infrastructure, it is also possible to engage with administrators about quantifying and offsetting the emissions produced, as well as improving the efficiency of your grid – this may help bring your organization toward carbon neutrality and have a significant impact at scale.

Select Data Center Location While many cloud providers are carbon neutral, some of their data centers may still be carbon intensive due to the local grid that they are connected to, whereas others will be low carbon and powered solely by renewable energy sources. Hence, selecting the data center location where an algorithm will be trained has a large impact on its direct carbon emissions. This choice can be achieved by consciously selecting the server location before dispatching your jobs. As we illustrated in previous sections, this single choice can make the direct emissions of an algorithm vary by a factor of 40, from 20g CO_2eq/kWh in a location that uses renewable energy sources to 820g CO_2eq/kWh in a location that solely relies on fossil fuels [5]. For a model such as VGG [18] or BERT [19], which are trained on multiple GPUs for several weeks, this can correspond to avoiding emitting several hundreds of kilograms of CO_2eq by training on a server powered by hydroelectricity instead of fossil fuels.

Reduce Wasted Ressources Grid search is still often used in practice, in spite of its low efficiency both in terms of model performance and environmental impact. However, it has been shown that random search (and others) not only is a straightforward replacement but also has potential to significantly accelerate hyperparameter search [20, 21, 22], consequently reducing carbon emissions. Also, while failed experiments are a common part of ML research and are sometimes unavoidable, their number can often be reduced with careful design such as unit tests, integration tests, and extensive and early debugging. Uninformative experiments are also frequent (sometimes unknowingly) – they can be caused by unstable learning algorithms requiring averaging results over many random seeds. Taking the time to carry out a literature review and to understand the potential sources of noise before launching large-scale hyperparameter searches increases the chance of obtaining reproducible and statistically significant results. Hence, reducing the need to extend the experiment cycles.

Choose More Efficient Hardware The choice of computing hardware can also have a major impact on ML emissions. To perform a comparison between different devices, their compute efficiency can be estimated in FLOPS/W. This estimation is based on their theoretical peak performance with respect to their Thermal Design Power (TDP)[3]. Using this approach, it can be found that CPUs can be 10 times less efficient than GPUs while TPU 3 can be 4 to 8 times more efficient than GPUs [23] (refer to Table 4 for details). Interestingly, in contexts where low power consumption and efficiency are important, e.g., for embedded applications, GPUs such as the Jetson AGX Xavier can be 10 to 20 times more efficient than traditional GPUs.

4 Discussion

The factors that we discussed in the current work give ML practitioners a certain amount of control over the environmental impact produced by the training of their models. We are aware that these choices are not always possible to make in practice – for instance, the choice of server location can be limited due to privacy considerations in the case of applications in the medical or financial domain, and large amounts of data may be needed to produce most robust models. However, we find that our emissions calculator is a good starting point to estimate the impact that small choices in model training can have on direct carbon emissions resulting from ML research.

Despite our best efforts, our calculator remains simply an approximation of the true emissions produced by ML training for several reasons: to start with, there is the issue of global load balancing, i.e. if a majority of practitioners choose to run their models in a low-carbon location, the servers will get saturated and other servers will still need be used. In that perspective, the global gain will not be a 40-fold reduction of emissions, but much smaller. Furthermore, there is a lack of transparency with regards to the true quantity of emissions produced by organizations, so while we use the current best publicly-available sources, there is still a large margin of error with regards to the exact quantity of energy consumed and carbon produced – we remain open to additional data sources and numbers. Finally, while in the current version of our tool, we focus on quantifying the emissions of training ML models, there is still the issue of deploying them, since the inference process is also energy-expensive, especially if done continuously and on a large scale. This is something that should be taken into account by ML practitioners in their products that are deployed in real-world settings, for instance by using energy-efficient architectures [24] and computing infrastructure.

There are also more far-reaching discussions to be had regarding the environmental value of scientific knowledge in general and of ML research in particular. On one hand, there is valuable research to be done in ML especially with regards to tackling climate change [25, 26], whereas on the other hand, the emissions of the field of ML are growing quickly [1]. We do not propose the solution to this problem, but we believe that there are steps to be taken, for instance by using efficiency as an evaluation criterion (as proposed by [2]) or by taking concrete steps to reduce emissions (as proposed by the current paper). We hope that our work, along with others, will open the door for these conversations and debates to take place, to quantify the environmental impact of our field, and for positive changes that can be made to reduce it.

[3]Empirical measurement of GFLOPS/W on various ML architecture would provide more accurate numbers but we are only interested in approximate values to compare classes of devices.

References

[1] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

[2] Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. Green ai. *arXiv preprint arXiv:1907.10597*, 2019.

[3] Simon Eggleston, Leandro Buendia, Kyoko Miwa, Todd Ngara, and Kiyoto Tanabe. *2006 IPCC guidelines for national greenhouse gas inventories*, volume 5. Institute for Global Environmental Strategies Hayama, Japan, 2006.

[4] WM To and Peter KC Lee. Ghg emissions from electricity consumption: A case study of hong kong from 2002 to 2015 and trends to 2030. *Journal of cleaner production*, 165:589–598, 2017.

[5] Matthew Brander, Aman Sood, Charlotte Wylie, Amy Haughton, and Jessica Lovell. Electricity-specific emission factors for grid electricity. *Ecometrica, Emissionfactors. com*, 2011.

[6] Ari Harju, Topi Siro, Filippo Federici Canova, Samuli Hakala, and Teemu Rantalaiho. Computational physics on graphics processing units. In *Proceedings of the 11th international conference on Applied Parallel and Scientific Computing*, pages 3–26. Springer-Verlag, 2012.

[7] Nima Tajbakhsh, Jae Y Shin, Suryakanth R Gurudu, R Todd Hurst, Christopher B Kendall, Michael B Gotway, and Jianming Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE transactions on medical imaging*, 35(5):1299–1312, 2016.

[8] Keiji Yanai and Yoshiyuki Kawano. Food image recognition using deep convolutional network with pre-training and fine-tuning. In *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6. IEEE, 2015.

[9] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[10] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[11] Stefan Falkner, Aaron Klein, and Frank Hutter. Bohb: Robust and efficient hyperparameter optimization at scale. *arXiv preprint arXiv:1807.01774*, 2018.

[12] Matthias Feurer and Frank Hutter. Hyperparameter optimization. In *Automated Machine Learning*, pages 3–33. Springer, 2019.

[13] Google. Google environmental report 2018, 2018.

[14] Microsoft. Beyond carbon neutral. white paper, 2018.

[15] Amazon Web Services. Aws & sustainability, 2019.

[16] Jim Gao. Machine learning applications for data center optimization, 2014.

[17] Google Data Centers efficiency: How we do it. `https://www.google.com/about/datacenters/efficiency/internal/`, 2019. Accessed: 2019-08-23.

[18] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[20] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *arXiv preprint arXiv:1603.06560*, 2016.

[21] Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Moritz Hardt, Benjamin Recht, and Ameet Talwalkar. Massively parallel hyperparameter tuning. *arXiv preprint arXiv:1810.05934*, 2018.

[22] Stefan Falkner, Aaron Klein, and Frank Hutter. Bohb: Robust and efficient hyperparameter optimization at scale. *arXiv preprint arXiv:1807.01774*, 2018.

[23] Paul Teich. Tearing apart google's tpu 3.0 ai coprocessor. `https://www.nextplatform.com/2018/05/10/tearing-apart-googles-tpu-3-0-ai-coprocessor/`, 2018.

[24] Ermao Cai, Da-Cheng Juan, Dimitrios Stamoulis, and Diana Marculescu. Neuralpower: Predict and deploy energy-efficient convolutional neural networks. *arXiv preprint arXiv:1710.05420*, 2017.

[25] David Rolnick, Priya L Donti, Lynn H Kaack, Kelly Kochanski, Alexandre Lacoste, Kris Sankaran, Andrew Slavin Ross, Nikola Milojevic-Dupont, Natasha Jaques, Anna Waldman-Brown, et al. Tackling climate change with machine learning. *arXiv preprint arXiv:1906.05433*, 2019.

[26] Victor Schmidt, Alexandra Luccioni, S. Karthik Mukkavilli, Narmada Balasooriya, Kris Sankaran, Jennifer Chayes, and Yoshua Bengio. Visualizing the consequences of climate change using cycle-consistent adversarial networks. *CoRR*, abs/1905.03709, 2019.

Appendix A: Energy Grid Data Used for the ML Emissions Calculator

For clarity purposes, the data presented in this appendix contains fewer columns than what can be found in our public database: https://github.com/mlco2/impact/tree/master/data. For instance we did not include the sources in these tables. For future reference, the data's commit hash at the time of publication is e692e28.

Google Cloud Platform

Region	Country	City	Estimated gCO2e/kWh)
asia-east1	Taiwan	Changhua County	557
asia-east2	China	Hong Kong	702
asia-northeast1	Japan	Tokyo	516
asia-northeast2	Japan	Osaka	516
asia-south1	India	Mumbai	920
asia-southeast1	Singapore	Jurong West	419
australia-southeast1	Australia	Sydney	802
europe-north1	Finland	Hamina	211
europe-west1	Belgium	St. Ghislain	267
europe-west2	United Kingdom	London	623
europe-west3	Germany	Frankfurt	615
europe-west4	Netherlands	Eemshaven	569
europe-west6	Switzerland	Zürich	16
northamerica-northeast1	Canada	Montréal	20
southamerica-east1	Brazil	São Paulo	205
us-central1	USA	Council Bluffs	566.3
us-east1	USA	Moncks Corner	367.8
us-east4	USA	Ashburn	367.8
us-west1	USA	The Dalles	297.6
us-west2	USA	Los Angeles	240.6

Amazon Web Services

Region	Country	City	gCO2e/kWh
us-east-2	USA	Columbus	568.2
us-east-1	USA	Ashburn	367.8
us-west-1	USA	San Francisco	240.6
us-west-2	USA	Portland	297.6
ap-east-1	China	Hong Kong	702
ap-south-1	India	Mumbai	920
ap-northeast-3	Japan	Osaka	516
ap-northeast-2	South Korea	Seoul	517
ap-southeast-1	Singapore	Singapore	419
ap-southeast-2	Australia	Sydney	802
ap-northeast-1	Japan	Tokyo	516
ca-central-1	Canada	Montreal	20
cn-north-1	China	Beijing	680
cn-northwest-1	China	Zhongwei	680
eu-central-1	Germany	Frankfurt am Main	615
eu-west-1	Ireland	Dublin	617
eu-west-2	United Kingdom	London	623
eu-west-3	France	Paris	105
eu-north-1	Sweden	Stockholm	47
sa-east-1	Brazil	Sao Paulo	205
us-gov-east-1	USA	Dublin	568.2
us-gov-west-1	USA	Seattle	297.6

Microsoft Azure

Region	Country	City	gCO2e/kWh
eastasia	Hong Kong	Wan Chai	702
southeastasia	Singapore	Singapore	419
centralus	USA	Des Moines	736.6
eastus	USA	Blue Ridge	367.8
eastus2	USA	Boydton	367.8
westus	USA	San Francisco	240.6
northcentralus	USA	Chicago	568.2
southcentralus	USA	San Antonio	460.4
northeurope	Ireland	Dublin	617
westeurope	Netherlands	Amsterdam	569
japanwest	Japan	Osaka-shi	516
japaneast	Japan	Tokyo	516
brazilsouth	Brazil	Sao Paulo	205
australiaeast	Australia	Sydney	802
australiasoutheast	Australia	Melbourne	805
southindia	India	Pallavaram	920
centralindia	India	Lohogaon	920
westindia	India	Mumbai	920
canadacentral	Canada	Toronto	69.3
canadaeast	Canada	Quebec	20
uksouth	United Kingdom	Midhurst	623
ukwest	United Kingdom	Wallasey	623
westcentralus	USA	Mountain View	297.6
westus2	USA	Quincy	297.6
koreacentral	South Korea	Seoul	517
koreasouth	South Korea	Busan	517
francecentral	France	Huriel	105
francesouth	France	Realmont	105
australiacentral	Australia	Forrest	900
australiacentral2	Australia	Forrest	900
southafricanorth	South Africa	Pretoria	1009
southafricawest	South Africa	Stellenbosch	1009

Appendix B: Hardware Efficiency

Name	Watt (TDP)	TFLOPS32	TFLOPS16	GFLOPS32/W	GFLOPS16/W
RTX 2080 Ti	250	13.45	26.90	53.80	107.60
RTX 2080	215	10.00	20.00	46.51	93.02
GTX 1080 Ti	250	11.34	0.17	45.36	0.68
GTX 1080	180	8.00	0.13	44.44	0.72
AMD RX480	150	5.80	5.80	38.67	38.67
Titan V	250	14.90	29.80	59.60	119.20
Tesla V100	300	15.00	30.00	50.00	100.00
TPU2	250	22.00	45.00	88.00	180.00
TPU3	200	45.00	90.00	225.00	450.00
Intel Xeon E5-2699	145	0.70	0.70	4.83	4.83
AGX Xavier	30	16.00	32.00	533.33	1066.67