An Analysis of Energy Requirement for Computer Vision Algorithms

Daniel Edelman
MIT Lincoln Laboratory
Supercomputing Center
Cambridge, MA
danieled@mit.edu

Siddharth Samsi
MIT Lincoln Laboratory
Supercomputing Center
Cambridge, MA
sid@ll.mit.edu

Joseph McDonald
MIT Lincoln Laboratory
Supercomputing Center
Cambridge, MA
jpmcd@ll.mit.edu

Adam Michaleas
MIT Lincoln Laboratory
Supercomputing Center
Cambridge, MA
Adam.Michaleas@ll.mit.edu

Vijay Gadepally
MIT Lincoln Laboratory
Supercomputing Center
Cambridge, MA
vijayg@ll.mit.edu

bstract—**The energy requirements of neural network learning are growing at a rapid rate. Increased energy demands have caused a global need to seek ways to improve energy efficiency of neural network learning. This paper aims to establish a baseline on how adjusting basic parameters can affect energy consumption in neural network learning on Computer Vision tasks. In this article, we catalog the effects of various adjustments, from simple batch size adjustments to more complicated hardware settings (e.g., power capping). Based on our characterizations, we have found numerous avenues to adjust computer vision algorithm energy expenditure. For example, switching from a single precision model to mixed precision training can result in energy reductions of nearly 40%. dditionally, power capping the Graphical Processing Unit (GPU) can reduce energy cost by an additional 10%.**

I. Introduction

The computational cost of machine learning is rising at an incredibly rapid rate. In an effort to improve performance of models, one can simply throw additional computational power at the problem for minor improvements. Computation Costs nearly double every 3–4 months [1], [2]. Part of this is caused by the development of larger and larger neural networks that have more parameters [3] and require increased training flops count (visualized in Figure 1, [4]) This trend is widely understood and deeply concerning [5]–[7]. This increase in computational power is having a very poor effect on the environment, as more and more energy has to be devoted to training these models for minimal improvement (the relationship between performance and compute is logarithmic, requiring exponentially more compute time for linear increases in performance [8]). Deep Learning Datacenters currently consume 1% of global energy reserves, and this value is expected to rise to 8–21% by the year 2030 [9], [10] and cost between 200 and 500 million dollars [11]. There is a great need to research and analyze the energy efficiency and performance of neural network learning, as well as create methods to develop better practices and standards to focus on performing efficient machine network learning that aims to reduce the carbon impact of deep learning [10], [12], [13].

Our current climate crisis is directly related to global energy demands and problems created from the use of nonrenewable energy sources to meet our energy needs. Massive overhauls to our energy habits are needed to stay under the 1.5 degree rise in global temperature before catastrophic climate effect could occur [14]. We need to investigate and overhaul how we produce and consume energy in order to limit global consequences.

This paper analyzes and catalogs how some simple adjustments can affect performance of several Computer Vision tasks. The goal is to provide a series of benchmarks one can use to catalog future performance and use as a stepping stone for future research on improving the energy efficiency of computer vision learning.

II. Related Work

One major problem with current machine learning analysis practices is that most are intensively performance focused [15]. A new architecture and design may show improved performance and meet a higher accuracy, but the increased training time and costs are not mentioned. Since the primary measurement is performance, other features (including energy costs, algorithmic efficiency) are ignored in pursuit of a singular metric. Since the simplest way to increase performance is to throw additional computing power at the problem, this highly promotes inefficiency and poor practices and heavily

This material is based upon work supported by the Assistant Secretary of Defense for Research and Engineering under Air Force Contract No. FA8702-15-D-0001, and United States Air Force Research Laboratory Cooperative Agreement Number FA8750-19-2-1000. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Assistant Secretary of Defense for Research and Engineering, or the United States Air Force. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

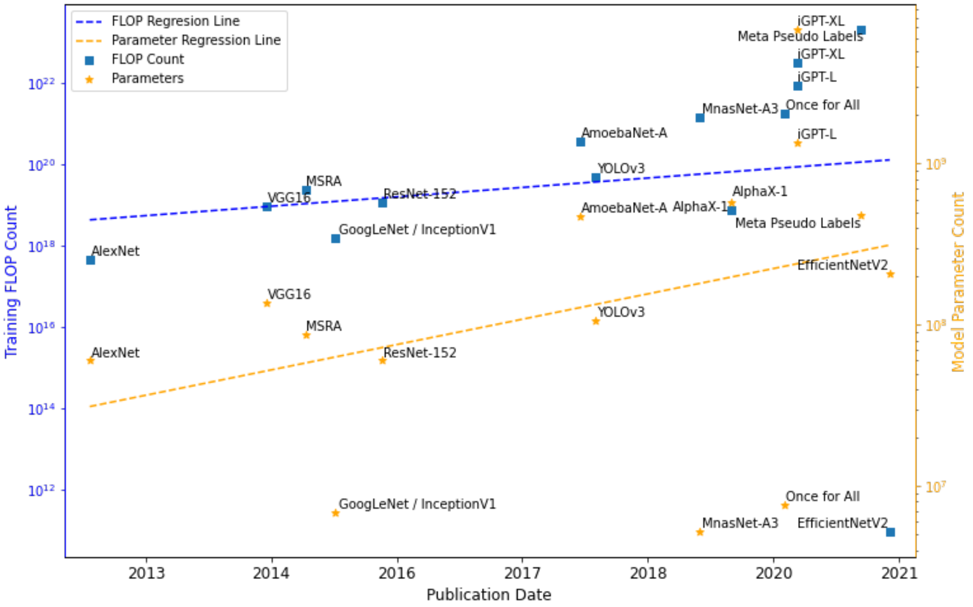


Fig. 1: Parameter and Flop Count of Computer Vision Neural Networks [4]

biases results in favor of large organizations that can afford large hardware setups [12].

. *Reducing Energy for I training such NLP Computer Vision*

Some research has been done to catalog the expense of Natural Language Processing (NLP) models. Some NLP models can consume as much *O* emission as the lifetime emission of five cars [16]. This caused a collection of work that analyzes power consumption on NLP learning tasks [17], as well as the energy demand of this training. [16] This paper seeks to analyze computer vision tasks and, hopefully, apply the results found for NLP tasks to computer vision and find generally applicable results.

Additionally, one promising avenue of research has been the investigation of power capping Graphics Processing Unit (GPU) on NLP training [18]. Power capping involves limiting the maximum power that can flow through the GPU or Central Processing Unit (CPU) nodes and thus reducing electrical consumption. Power capping has shown limited progress in others tasks [19] and we would like to see if the results generalize to other fields of machine learning.

Existing research has explored how various architectures perform on ImageNet [20]. This indicates a linear relationship between the total operation required to run a model and total inference time. Additionally, this indicates that batch size has a minimal influence on energy consumption.

Existing work has been done in analyzing improvement and lowering the computation load. Designing architecture to minimize total flops and computational load [21] in order to speed up training without major losses in accuracy is necessary.

B. *Computer Vision Benchmarks*

Now we recall the current benchmarks established for computer vision tasks as well as the common metrics to analyze current performance. Existing literature on computer vision and other AI benchmarks measure the time needed to train the model. The goal is to create a set of standards and measure the total training time it takes to train a model to a desired accuracy. This allows testing of multiple algorithms and methods, to compare and contrast the accuracy and training time of the various approaches.

MLperf is one such collection of benchmarks for various neural network learning tasks from image processing to object segmentation [22]. It contains records for several tasks including object detection, image classification, natural language processing, and more. There have been many submissions from various organizations about how their own hardware and algorithms perform on the various tasks. Sadly, MLperf only records the total training time needed to reach a desired accuracy. There is little documentation on the energy requirement for an individual organization's submission. This work hopes to establish a starting point for energy demand in computer vision neural network training.

One issue with cataloging the energy cost of machine learning is the lack of a widely accepted framework to measure the climate impact of machine learning. However, attempts to create a standardized or consistent system to measure climate impact of AI learning are being developed [23]. There are many avenues of exploration for methods of measuring the energy demand and climate impact of machine learning.

Additionally, there is a growing trend to consider Green AI [10], [12], where environmental impact and practices are recorded. Instead of constantly seeking better performance, algorithmic efficiency is considered as well.

III. Analysis of Hyperparameter Selection

The ImageNet [24] dataset is a massive computer vision dataset that is used to train models for computer vision tasks. ImageNet consists of numerous images (over 1 million) of various categories. ImageNet is frequently used to train neural network models for image classification tasks. Image classification is when a model is given an image and needs to clarify what it is a picture of (e.g., predicting if an image was a cat or a dog). This is a very simple process for humans, but quite complicated for machines.

The chosen neural network architecture will be ResNet50 [25]. This architecture will be used for our experiments. ResNet50 (residual network) is a widely accepted model architecture for image classifications with ImageNet, and is even used as the accepted model architecture in the MLperf category for image classification [22]. The goal of each experiment will be to reach 70% top 1 accuracy on our ImageNet validation set, and the experiment will cease upon reaching this goal.

Additionally, we will be using the You Only Look Once (YOLO) architecture [26] on the xView dataset [27] to provide a sense of comparison to our finding on other computer vision tasks. YOLO is another neural network architecture designed around object detection (identifying where an image is in a picture, rather than what the image is a picture of). The xView dataset consist of overhead images take of location from the air to identify locations on the ground (buildings, vehicles).

We hope to get a sense of how adjusting these algorithmic features changes the energy consumption and training time. Additionally, we want to gain a sense of how much influence each adjustment has on the various metrics. Knowledge of how each parameter influences those metrics could allow us to find the optimal configuration for learning.

. *Batch Size*

One of the simplest of the hyperparameters, for neural network learning to adjust, is the batch size. Due to parallel processing, networks can process and evaluate multiple data points at once before each training update. The amount of data points being processed at once is the batch size. Larger batch sizes process more data and finish faster but consume larger amounts of memory and have greater memory overhead.

For ResNet and ImageNet we used a batch size of 64, 128, 256, and 512 in our experiments (with 256 being used as a reference point). For our YOLO experiments we used batch size of 32, 64, 128, 256, 512 as values to be used. We can see how this impacted energy consumption in Figure 2.

The batch size that reached the target accuracy in the fewest epochs was between 128 and 256. Further analysis (as seen in Figure 2) on the total time taken and energy consumed shows remarkable similarities in both energy consumption and time taken among these two batch sizes. However, analysis of the accuracy per epoch, shows that a batch size of 256 was slightly better. Consuming about 20% less energy than the other batch size 64 and 512.

Conversely, upon examining the performance with the YOLO architecture, no significant difference was seen. However, the xView dataset is much smaller than ImageNet and thus may be obscuring possible values or trends that can only be seen on a larger scale.

B. *Network Complexity*

For a given neural network, there is an underlying philosophy in how the network is structured and created. For example, ResNet architectures use residual layers [25] where the current storage is sent to the future. The various networks (e.g, ResNet18, ResNet34, ResNet50) use the same underlying design with a crucial difference in total number of layers. Increasing the complexity of the network and increasing the total number of parameters allows for more flexible learning and higher accuracy at the expense of greatly increasing the computational cost [5].

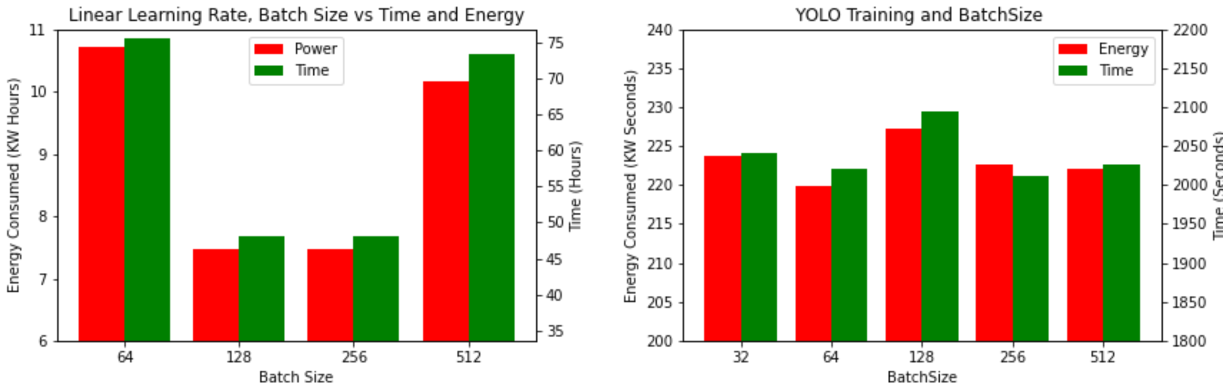
We utilized both the ImageNet dataset and xView dataset to train various models of ResNet (on ImageNet) and YOLO (on xView). We measured performance both under several epochs of training (40 epochs for ResNet and 100 epochs on YOLO) as well as a single inference pass. We can see the results in Figure 3 and Figure 4 for ResNet and YOLO respectively.

As can be seen, as the complexity of the network is increased, we see increases in energy consumption as well. The increase in parameters results in an increase in the required energy consumption for training processes. The desired tradeoff can be seen in Figure 5. We compare the top 1 Accuracy of our models with the training time and energy consumption and see a sharp decrease in diminished returns, where a small increase in accuracy requires a much larger energy consumption.

Utilizing ResNet50 as a baseline, we see that increasing the complexity to ResNet101 causes a near 50% increase in energy consumption while a further increase to ResNet152 nearly doubles energy consumption; lowering the complexity can shave energy consumption by 20%. Therefore, we see that increasing the parameter count results in higher accuracy with sharply diminishing returns in the final performance.

C. *Precision*

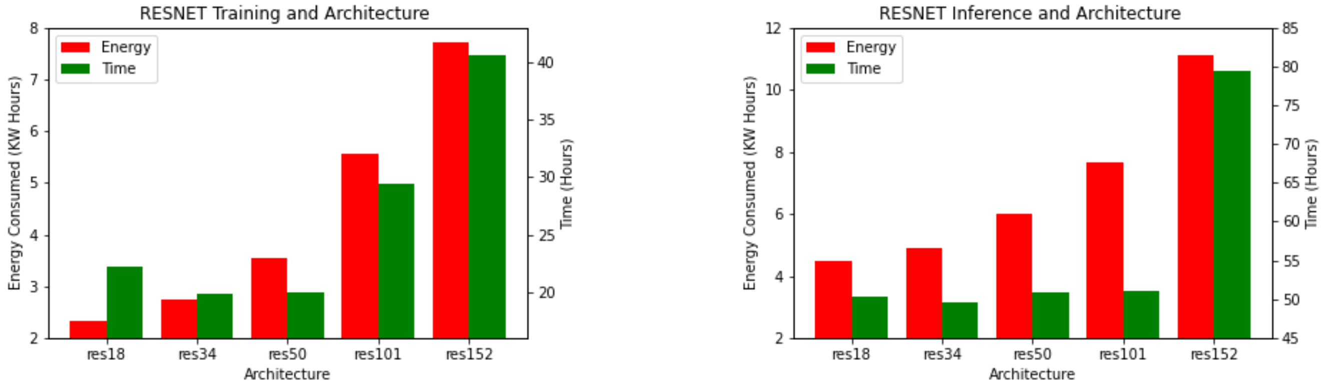
When calculating and performing operation on a machine, there is some machine error performed with calculation. Since machines must use a fixed amount of bytes to store values, not every value can be represented and some small error will exist. This error is called "machine error." The error can be reduced by allowing for more bytes to store values; however, this increases the complexity of operations. The common levels of precision are single precision (or float32 which uses 32 bits to store numbers) or double precision (or float64 which uses 64



(a) RESNET

(b) YOLO

Fig. 2: Batch Sizes and performance



(a) Training

(b) Inference

Fig. 3: ResNet Architecture, training, and inference

bits to store numbers). As the name suggests, double precision has smaller errors than single precision but requires twice the computation to calculate operations with. (In theory, precision could be increased arbitrarily, but in most cases these levels should suffice for the task at hand). Now we will examine how precision effects energy usage.

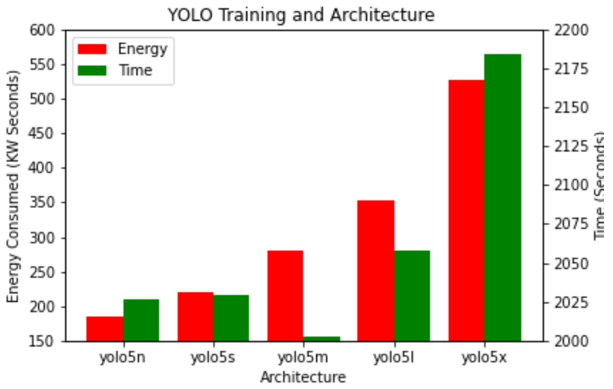
Most training in PyTorch uses float32 or single precision to store the values of the network. Both the values from the network and the loss from the forward pass are calculated using float32 values. However, there is an idea to save on possible computation by using a mixed precision model. In a mixed precision model, the forward pass is calculated using float16 bits, (and then the loss is scaled to prevent any adjustment errors from float16 to float32 for the backwards propagation). Since a lot of work is spent doing the forward pass for batches, this method could save a lot of computations and result in much increased efficiency as the GPU could process the forward pass twice as fast (as values as now operations are consuming half as many bits).

We will experiment with both a single and mixed precision model to identify the savings of improvements. Hopefully, by reducing the complexity of the operations during training, we can reduce the total computation time and total energy expenditure.

As can be seen in Figure 6, switching to mixed precision can improve and reduce energy consumption by nearly 40%. Additionally, the experimental result showed no significant change in model accuracy or performance on the data. Changing the precision seems to result in much faster and more efficient training, with minimal impact on performance and accuracy. This is a huge improvement and allows us to achieve massive more leverage in our computations by reducing complexity. If possible, experimenting with smaller precision (float8 or lower) could be investigated to see if further reduction in cost could be obtained.

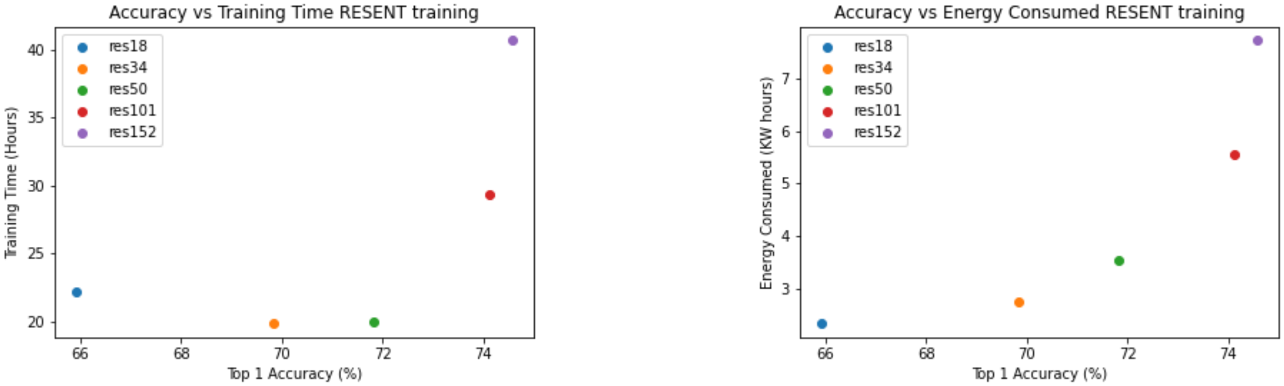
IV. Hardware Tuning

Neural network learning involves performing lots of matrix operations during training. The ability to perform these calculations in parallel is a task where the GPU excels.



(a) Training

Fig. 4: YOLO Architecture, training, and inference



(a) Time

(b) Energy Consumption vs. Accuracy

Fig. 5: ResNet Architecture and Performance

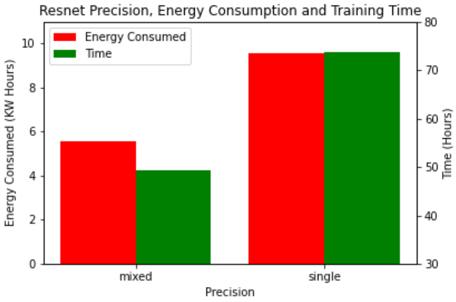


Fig. 6: Energy Consumption and Training Time of Single and Mixed Precision Models

Modern machine learning algorithms leverage this feature of the GPU to accelerate the training process [28]. These techniques heavily reduce total training time for a much faster and more efficient performance.

. *GPU Power Capping*

By doing many calculations quickly, the GPU plays a major and important role during neural network training. Therefore, analyzing the efficiency and performance of the GPU itself can be a possible method to identify ways to improve the energy efficiency of neural network learning.

While the GPU is being used for training, it is not always using 100% of its power as there is downtime between batches because the data needs to be loaded onto and from the GPU. Potentially, limiting the maximum power draw is one possible way of reducing energy. While in theory this will increase the time it takes because it must operate slower, results on NLP tasks have shown promise, and we wish to see if results replicate with computer vision tasks [17]. We will explore the impact and effect power capping the GPU will have on computer vision tasks in the following section.

One avenue of exploration was power capping the GPU to reduce the total power expenditure. The idea was to limit how much power the GPU can consume during training and hopefully alter and find improvements. Lowering how much energy the GPU can consume will hopefully reduce power demands of the network during training.

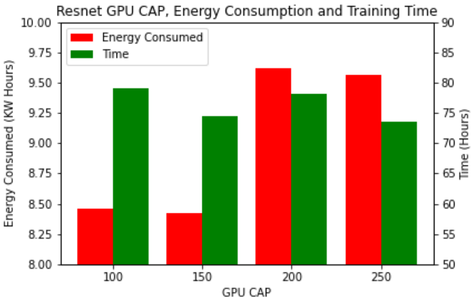


Fig. 7: Power Cap and Values

Experiments were set up with a GPU power cap of 250W (default), as well as reduction in the cap to 200W, 150W, and 100W. The idea was to measure the effect of reducing the power cap, as well as seeing if greater reduction resulted in greater improvements.

We can see in Figure 7 that lowering the power cap from 250W to 100W or 150W resulted in a savings of about 10%. We also note that the change and increase in run time was not significant enough to counteract these effects.

As it stands, lowering the power cap seems to result in reduced energy consumption without a significant drop in training time or accuracy, and thus could easily be applied in future data center learning by adjusting the default power cap of GPU. It is an easy and simple change that could result in massive energy savings.

V. Discussion

Through our course of experiments, we found that even simple adjustments (e.g., adjusting the batch-size) can influence how much energy was consumed during training and resulting impacts on efficiency. We catalog how much increasing the complexity of a model influences on performance and how it results in minor improvements in performance (a nearly 50% increase in energy consumption for around 3% increase in top 1 accuracy from jumping from ResNet50 to ResNet101, with even further diminishing returns going to ResNet152). We noticed massive gains and reduction in energy from switching from the single precision of nearly 40%. Showcasing how this makes neural network training more efficient by a massive value. Simple hardware adjustments – capping the GPU – resulted in a savings of 10% which when applied on a massive datacenter could have massive and incredible impacts on the energy impact of the datacenter. This implies that a datacenter can simply alter the default power cap of the GPU associated with jobs to result in major savings with minimal performance impact on the user. Such a simple change could result in a massive reduction in energy consumption.

VI. Conclusion

Overall, the results are a great starting point for future analysis and behavior. However, a lot more work could and will be done. Due to limitations, only a single model's GPU was available on our datacenter; therefore, we were unable to measure how various GPU models consume energy.

Since training and jobs involve both CPU and GPUs, one possible avenue of future exploration is to analyze if power-capping the CPU can also have a similar effect. Since CPUs are used for all datacenter job tasks (while GPUs are used for those that require parallel computation) finding out how power-capping the CPU influence performance can also be a major exploration to reduce datacenter electrical demand.

All in all, this is a promising future avenue of research that aims to reduce energy demand during neural network learning. Additional data on datacenter operation as well as non-AI learning tasks could also help find optimal settings for a datacenter to reduce energy demand while minimizing inconvenience for users.

VII. Acknowledgments

The authors acknowledge the MIT SuperCloud [29] and Lincoln Laboratory Supercomputing Center for providing HPC and consultation resources that have contributed to the research results reported within this paper. The authors acknowledge the MIT SuperCloud team: William Arcand, David Bestor, William Bergeron, Chansup Byun, Matthew Hubbell, Michael Houle, Mike Jones, Jeremy Kepner, Anna Klein, Judy Marchese, Peter Michaleas, Lauren Milechin, Julie Mullen, Andrew Prout, Albert Reuther, Antonio Rosa, and Charles Yee. The authors also wish to acknowledge the following individuals for their contributions and support: Bob Bond, Allan Vanterpool, Tucker Hamilton, Jeff Gottschalk, Tim Kraska, Mike Kanaan, Charles Leiserson, Dave Martinez, John Radovan, Steve Rejto, Daniela Rus, Marc Zissman.

References

[1] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M. M. A. Patwary, Y. Yang, and Y. Zhou, "Deep learning scaling is predictable, empirically," 2017.

[2] J. Sevilla, L. Heim, A. Ho, T. Besiroglu, M. Hobbhahn, and P. Villalobos, "Compute trends across three eras of machine learning," 2022.

[3] R. B. Roy, T. Patel, V. Gadepally, and D. Tiwari, "Bliss: auto-tuning complex applications using a pool of diverse lightweight learning models," in *Proceedings of the 42nd CM SIGPL N International Conference on Programming Language Design and Implementation*, 2021, pp. 1280–1295.

[4] J. Sevilla, P. Villalobos, J. F. Cer n, M. Burtell, L. Heim, A. B. Nanjajjar, A. Ho, T. Besiroglu, M. Hobbhahn, J.-S. Denain, and O. Dudney, "Parameter, compute and data trends in machine learning," https://docs.google.com/spreadsheets/d/1AAIebjNsnJj_uKALHbXNfn3_YsT6sHXtCU0q7OIPuc4/, 2022.

[5] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, "The computational limits of deep learning," 2020. [Online]. Available: https://arxiv.org/abs/2007.05558

[6] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green ai," *Communications of the CM*, vol. 63, no. 12, pp. 54–63, 2020.

[7] P. Dhar, "The carbon impact of artificial intelligence." *Nat. Mach. Intell.*, vol. 2, no. 8, pp. 423–425, 2020.

[8] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," 2017.

[9] D. Zhao, N. C. Frey, J. McDonald, M. Hubbell, D. Bestor, M. Jones, A. Prout, V. Gadepally, and S. Samsi, "A green(er) world for a.i." in *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, may 2022. [Online]. Available: https://doi.org/10.1109\%2Fipdpsw55747.2022.00126

[10] B. Li, S. Samsi, V. Gadepally, and D. Tiwari, "Green carbon footprint for model inference serving via exploiting mixed-quality models and gpu partitioning," *arXiv preprint arXiv:2304.09781*, 2023.

[11] B. Cottier, "Trends in the dollar training cost of machine learning systems," 2023, accessed: 2023-7-2. [Online]. Available: https://epochai.org/blog/trends-in-the-dollar-training-cost-of-machine-learning-systems

[12] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green ai," 2019. [Online]. Available: https://arxiv.org/abs/1907.10597

[13] D. Edelman, "Characterizing the energy requirement of computer vision," Master's thesis, MIT, 2023.

[14] F. Harvey, "Scientists deliver 'final warning' on climate crisis: act now or it's too late," 3 2023. [Online]. Available: https://www.theguardian.com/environment/2023/mar/20/ipcc-climate-crisis-report-delivers-final-warning-on-15c

[15] D. Hernandez and T. B. Brown, "Measuring the algorithmic efficiency of neural networks," 2020.

[16] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in NLP," in *Proceedings of the 57th nnual Meeting of the ssociation for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 3645–3650. [Online]. Available: https://aclanthology.org/P19-1355

[17] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean, "Carbon emissions and large neural network training," 2021. [Online]. Available: https://arxiv.org/abs/2104.10350

[18] J. McDonald, B. Li, N. Frey, D. Tiwari, V. Gadepally, and S. Samsi, "Great power, great responsibility: Recommendations for reducing energy for training language models," 2022. [Online]. Available: https://arxiv.org/abs/2205.09646

[19] A. Haidar, H. Jagode, P. Vaccaro, A. YarKhan, S. Tomov, and J. Dongarra, "Investigating power capping toward energy-efficient scientific applications," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 6, p. e4485, 2019, e4485 cpe.4485. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.4485

[20] A. Canziani, A. Paszke, and E. Culurciello, "An analysis of deep neural network models for practical applications," 2016. [Online]. Available: https://arxiv.org/abs/1605.07678

[21] J. Chen, S. hong Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee, and S. H. G. Chan, "Run, don't walk: Chasing higher flops for faster neural networks," 2023.

[22] P. Mattson, C. Cheng, G. Diamos, C. Coleman, P. Micikevicius, D. Patterson, H. Tang, G.-Y. Wei, P. Bailis, V. Bittorf *et al.*, "Mlperf training benchmark," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 336–349, 2020.

[23] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, and J. Pineau, "Towards the systematic reporting of the energy and carbon footprints of machine learning," 2022.

[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[26] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, K. Michael, TaoXie, J. Fang, imyhxy, Lorna, Z. Yifu, C. Wong, A. V, D. Montes, Z. Wang, C. Fati, J. Nadar, Laughing, UnglvKitDe, V. Sonck, tkianai, yxNONG, P. Skalski, A. Hogan, D. Nair, M. Strobel, and M. Jain, "ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation," Nov. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.7347926

[27] D. Lam, R. Kuzma, K. McGee, S. Dooley, M. Laielli, M. Klaric, Y. Bulatov, and B. McCord, "xview: Objects in context in overhead imagery," 2018.

[28] B. Li, R. Arora, S. Samsi, T. Patel, W. Arcand, D. Bestor, C. Byun, R. B. Roy, B. Bergeron, J. Holodnak *et al.*, "Ai-enabling workloads on large-scale gpu-accelerated system: characterization, opportunities, and implications," in *2022 IEEE International Symposium on High-Performance Computer rchitecture (HPC)*. IEEE, 2022, pp. 1224–1237.

[29] A. Reuther, J. Kepner, C. Byun, S. Samsi, W. Arcand, D. Bestor, B. Bergeron, V. Gadepally, M. Houle, M. Hubbell, M. Jones, A. Klein, L. Milechin, J. Mullen, A. Prout, A. Rosa, C. Yee, and P. Michaleas, "Interactive supercomputing on 40,000 cores for machine learning and data analysis," in *2018 IEEE High Performance extreme Computing Conference (HPEC)*. IEEE, 2018, pp. 1–6.