

Energy Efficiency in Machine Learning: A position paper

Eva García-Martín, Niklas Lavesson, Håkan Grahn and Veselka Boeva
Department of Computer Science and Engineering
Blekinge Institute of Technology, 371 79, Karlskrona, Sweden
Email: {eva.garcia.martin, niklas.lavesson, hakan.grahn, veselka.boeva}@bth.se

Abstract

Machine learning algorithms are usually evaluated and developed in terms of predictive performance. Since these types of algorithms often run on large-scale data centers, they account for a significant share of the energy consumed in many countries. This position paper argues for the reasons why developing energy efficient machine learning algorithms is of great importance.

1 Introduction

Machine learning algorithms are becoming more and more popular due to the availability of large volumes of data and the advancements in hardware that makes it possible to analyze these data. These algorithms are present in large-scale data centers, which account for 3% of the global energy consumption¹. This energy consumption needs to be reduced due to health issues linked to environmental pollution [1]. There are two ways to address this challenge, either by researching on how to find new, clean, sources of energy that can provide enough energy for the population, or to reduce the actual energy consumption of our devices. We center on the second one: building sustainable and energy-efficient machine learning algorithms.

There is a lot of research conducted in machine learning focusing on improving the predictive performance of algorithms, but recently researchers are becoming more interested on improving energy efficiency as well. This paper argues for the reasons why developing energy efficient algorithms in machine learning is of great importance. We focus on three counterclaims to develop machine learning algorithms considering energy efficiency: i) reducing the energy consumption of machine learning algorithms does not necessarily lead to a reduction of the overall energy consumption, ii) time and energy are strongly correlated, thus being redundant to measure the energy consumption since time is already measured², and iii) it is complicated to measure energy consumption, thus making it time consuming and impractical [2].

The rest of the paper is organized as follows: Section 2 provides the background with the terminology and concepts related to time, energy, and power, together with the related works. Section 3 explains in detail the counterclaims for studying energy efficiency in machine learning. We address those counterclaims in Sections 4, 5 and 6. Section 7 portrays some preliminary results of having measured the energy consumption of a particular algorithm in different datasets. Finally, Section 8 presents the conclusions.

¹<http://www.independent.co.uk/environment/global-warming-data-centres-to-consume-three-times-as-much-energy-in-next-decade-experts-warn-a6830086.html>

²<https://01.org/powertop/overview>

2 Background

2.1 Terminology

All the equations and notation are based on the work by Dubois et.al [3]. Our scope centers on showing how energy is consumed by an algorithm, presenting the relationship between energy, time, and the instructions of a program. These instructions are to be optimized towards an overall energy reduction.

Energy (joules) is the product of power (watts) and time (seconds),

$$energy = power \cdot time \quad (1)$$

energy being the amount of power consumed during a period of time. Power is a measurement of the rate at which energy is consumed. Since

$$dynamic \quad power = C \cdot V_{dd}^2 \cdot f \quad (2)$$

V_{dd} being the voltage, C the capacitance and f the frequency, we observe that the relationship between time and energy is nonlinear. Lowering the frequency of the processor leads to longer executions, but the total energy consumption can be lower, since V_{dd} can be reduced at a lower frequency, thus lowering the power significantly and lowering the energy consumption.

The total execution time of a program (multicycle computer model) is the multiplication of the number of instructions (IC), the average number of clock cycles per instruction (CPI), and clock cycle time (TPC):

$$T = IC \cdot CPI \cdot TPC \quad (3)$$

The total energy consumed by a program is

$$E = IC \cdot CPI \cdot EPC \quad (4)$$

where EPC is the energy per clock, and it is defined by integrating the power over a period of time, thus,

$$EPC = \frac{1}{2} \cdot C \cdot V_{dd}^2 \quad (5)$$

$EPI = CPI \cdot EPC$, where EPI is energy per instruction. CPI represents the number of clocks needed to execute an instruction. The key is that CPI is different for the types of instructions, thus, there are certain instructions that need more CPI

than others. For instance, a load instruction has a higher CPI than an ALU instruction. A way to optimize a program to consume less energy and to take less time is to use more instructions with a lower CPI and less instructions with a higher CPI.

2.2 Related Work

Energy consumption has been widely studied for many years in the computer engineering community, focusing on designing processors that consume very little energy using Dynamic Voltage Frequency Scaling (DVFS) and on some other power saving techniques [4]. Regarding software, green computing has been introduced as a field that studies ways to address software solutions from a green, sustainable, and energy efficient perspective. Many companies are also starting to be concerned with the energy consumption of their computations³. Based on these works, we can observe how there is a trend in building energy efficient software [5, 6, 7].

In relation to machine learning, a concern for building energy efficient algorithms is increasing in the community. For instance, in a panel discussion during the 2016 Knowledge Discovery and Data Mining conference held in San Francisco⁴, there was a very interesting discussion between key researchers in the field addressing which steps should be taken to make algorithms more energy efficient. They centered in the deep learning field, since these algorithms are particularly computationally expensive [8]. Autonomous cars is a field that is directly connected to machine learning and energy efficiency. Autonomous cars often rely on deep learning algorithms, and since most of these cars are powered by batteries, it is important to build accurate models that consume lower amounts of energy than what is currently the case. There is already a research group making deep neural networks energy efficient while maintaining the same levels of accuracy [9].

3 Counterclaims

There are several aspects to consider related to developing energy efficient machine learning algorithms. The scope of this position paper centers in three counterclaims: i) reducing the energy consumption of machine learning algorithms does not necessarily lead to a reduction of the overall energy consumption, ii) time and energy are strongly correlated, thus being redundant to measure the energy consumption since time is already measured, and iii) it is complicated to measure energy consumption, thus making it time consuming and impractical.

Overall energy reduction

The first concern is that even if we spend tremendous efforts on both reducing the energy consumption of algorithms and studying machine learning from an energy efficient perspective, this does not necessarily lead to a significant reduction of energy. For instance, if the algorithms that we are studying are hardly used on large-scale platforms, optimizing their energy consumption will have no impact on the world's energy consumption. So the focus should be to reduce the energy in those algorithms that are being used daily, on large-scale data centers, and by different applications.

³<https://deepmind.com/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-40/>

⁴<http://www.kdd.org/kdd2016/>

Correlation between time and energy

The second counterclaim centers on the fact that in many cases energy and time can be so strongly correlated that measuring the energy consumption of algorithms is a waste of resources, since we already have that information from the execution time⁵. Adding the energy consumption variable to the algorithm design process might just create an overhead on the time to publish and release such an algorithm, since it is not so straightforward to measure energy consumption, which leads to the last counterclaim.

Measuring energy consumption

Measuring energy consumption can be quite troublesome due to the unavailability of tools that can give an accurate approximation of the energy [2]. Some tools are based solely on statistical modeling which are based on the usage of the CPU. Again, giving energy results that are correlated with time. Based on this, it can be unfeasible to try to understand how energy is consumed in an algorithm.

4 Impact on overall energy consumption

For the counterclaim: *reducing the energy consumption of machine learning algorithms does not necessarily lead to a reduction of the overall energy consumption* to be true, one of the following two claims need to be satisfied: i) that machine learning algorithms are hardly used in data centers, and that most of the energy consumption of those centers is due to other reasons, ii) that modifying these algorithms does not output a significant difference in terms of the energy consumed.

We can observe how many companies are building artificial intelligence (AI) solutions based on machine learning algorithms in their applications. Facebook has an AI research lab (FAIR)⁶ that is applying machine learning to enhance the user experience, with applications that go from face detection to search similar multimedia documents in a database. Google has several research divisions, such as Google Brain and Google Deep Mind, centered in machine learning. This pattern occurs in many other companies, Amazon, Yahoo, etc. Since machine learning solutions are present in these companies, which generate a high percentage of the total Internet traffic, we believe that machine learning algorithms are frequently being used in large-scale data centers.

Addressing the second point, we have already published some work that shows how energy consumption can vary depending on how an algorithm is programmed [10]. In Section 7 we show some results of running an algorithm in different datasets, and the difference in energy consumption.

With the increased interest that machine learning is having in the top companies which govern the Internet traffic, even a small reduction of the energy consumption of an algorithm will have a great impact on the overall energy consumption at a global scale.

⁵<https://01.org/powertop/overview>

⁶<https://research.fb.com/category/facebook-ai-research-fair/>

5 Correlation between energy and time

From Eq. 1 we can see that time and energy are directly connected through power. While time gives an overview of the efficiency of a computation, energy gives a more detailed overview into this matter. Measuring energy consumption can be a complement to measuring only time. If the execution of algorithm *A* is longer than the execution of algorithm *B*, that does not necessarily lead to a higher energy consumption by *A* compared to *B*. These algorithms could run in a cellphone, and the battery consumed by *B* could be higher. The reasons behind this is Eq. 2, and the main reasons behind DVFS. Reducing the frequency of the processor for a specific process can reduce V_{dd} , thus significantly reducing the power. While reducing the frequency might make the process run for longer, the total energy consumed could be less.

6 Measuring energy consumption

At the moment it is not straightforward how to measure the energy consumption of software. Some researchers have built energy models for specific algorithms [9] and others use statistical tools based on the CPU usage [11].

We proposed a methodology to measure the energy consumption at a fine-grained level of machine learning algorithms in a previous publication [12]. In such publication we used a tool based on statistical models. In this section we focus on describing how to measure the energy consumption with Sniper [13] for each function of the algorithm. We focus on analyzing the energy consumption of each function to understand where exactly is the energy hotspot of the algorithm.

Sniper⁷ is an x86 simulator that can run an algorithm and, with the use of McPAT⁸ (integrated in Sniper), output the energy consumed by the algorithm giving an overview of the instructions responsible for that. Some examples are given in Section 7.

In order to measure the energy consumption with sniper, we need to inject the code of the algorithm with markers around those functions or regions of interest where we would like a detailed energy measurement⁹. This is done by including `SimMarker()` calls around those functions. The algorithm is then compiled, and inserted in the sniper run, with the `roi` (region of interest) option activated, so that it takes into considerations the markers, saving statistics at every step. Once the execution is finished, we get an overview of the energy and power consumption by calling McPAT, using a script provided by sniper, and passing the different markers of the functions as parameters. In order to get the energy consumption of every function call, we need to have a different marker name for each call, otherwise we will only be able to extract the energy consumption from one function call.

7 Experiment

This section shows the energy consumption of three different runs of the same algorithm under three different datasets. The goal is to show how can energy vary depending on the type of execution, and the type of results obtained from measuring energy consumption with sniper.

⁷http://snipersim.org/w/The_Sniper_Multi-Core_Simulator

⁸<http://www.hpl.hp.com/research/mcpat/>

⁹http://snipersim.org/w/Multiple_regions_of_interest

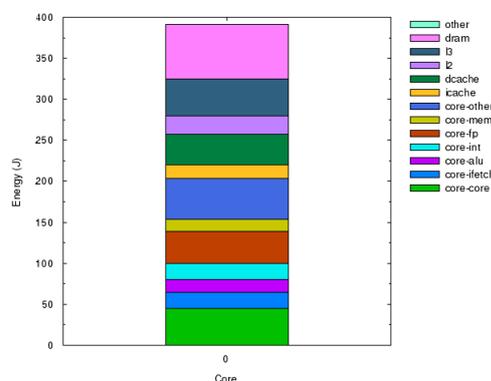


Figure 1: Energy consumption of the VFDT. Dataset = 1 million instances, 5 numeric and 5 nominal attributes.

7.1 Algorithm

The algorithm profiled is the Very Fast Decision Tree (VFDT) [14]. VFDT is a decision tree algorithm able to analyze data from a stream, in an online fashion, thus updating the model as the data arrives, not saving any data and reading it only once. It outputs competitive predictive performance results in comparison to standard offline decision tree algorithms, while being able to handle large amount of data. VFDT reads each instance once by once, and then updates the statistics of the attributes and the classes seen at each node. Once enough examples are seen at a specific node, if there is a clear attribute that has a higher information gain in comparison to the others, that attribute becomes a new node on the tree, and a split is made. This process is repeated until there is not more data in the stream.

7.2 Datasets

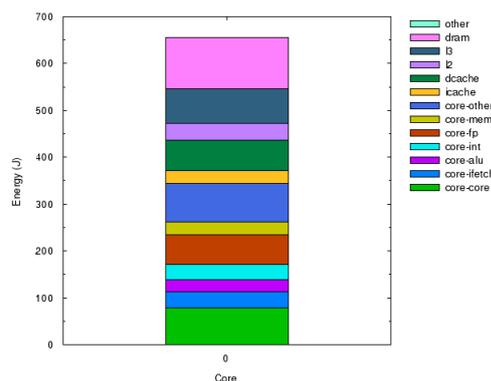


Figure 2: Energy consumption of the VFDT. Dataset = 100k instances, 50 numeric, 50 nominal attributes.

In order to get a general overview of the energy consumed by the VFDT, we run the algorithm under three different datasets. The first one had 1,000,000 instances, 5 numeric attributes and 5 nominal attributes. The second one had 100,000 instances,

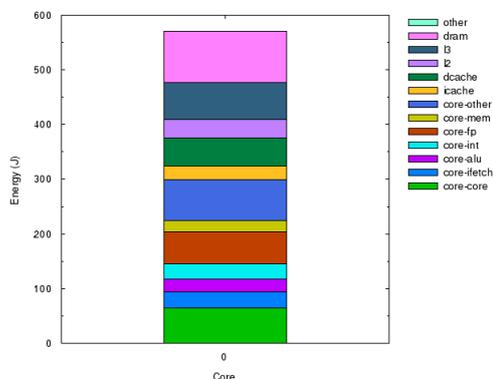


Figure 3: Energy consumption of the VFDT. Dataset = 100 thousand instances, 50 numeric attributes.

50 numeric attributes and 50 nominal attributes. The third one had 100,000 instances and 50 numeric attributes.

The goal was to test how energy varied in datasets with different number of attributes and instances, to try to generalize the energy consumption behavior of the algorithm. Keeping track of numeric attributes can be very troublesome, usually needing more energy than for nominal attributes. VFDTc was an update made to VFDT that could handle numerical attributes [15].

7.3 Results

Table 1 shows the energy consumed by running VFDT on the three datasets. We observe that there is a significant difference between the energy consumed from the three datasets. Interestingly, although D2 and D3 contain less instances than D1, they consume more energy, since they have more attributes. From the results, we can see that having more attributes consumes more energy than having more instances. D2 has 10 times less instances and 10 times more attributes, and it consumes 1.67 times more energy. D3 shows how expensive it is to analyze numeric attributes. Since basically, the only difference from D2 and D3 are 50 nominal attributes, and the difference is 85.16 J. Adding 50 nominal attributes is only 85.16 J, while the rest is spent on reading the data and analyzing those numeric attributes. One possible way to optimize VFDT in terms of energy consumption is to see which method is best to handle numeric attributes [16].

Figures 1, 2, and 3 show the total energy consumed in those three executions, specifying which instructions are consuming that energy. We can observe that they follow similar patterns, and that most of the energy is consumed due to RAM and cache accesses.

8 Conclusion and Future Work

This position paper argues for the importance of developing energy efficient machine learning algorithms. We address three counterclaims: i) reducing the energy consumption of machine learning algorithms does not necessarily lead to a reduction of the overall energy consumption, ii) time and energy are strongly correlated, thus being redundant to measure the energy consumption since time is already measured, and iii) it

Table 1: Results from executing the VFDT on three setups, generated with the random tree synthetic generator [17] Nom=Nominal attributes, Num=Numeric attributes.

Setups	Instances	Nom	Num	Energy (J)
S1	1,000,000	5	5	391.48
S2	100,000	50	50	656.26
S3	100,000	0	50	571.10

is complicated to measure energy consumption, thus making it time consuming and impractical.

We argue that reducing the energy consumption of machine learning algorithms will have an impact on the overall energy consumption, since the top companies responsible for most of the Internet traffic are using these type of algorithms. Examples are: Facebook’s AI research lab, and Google Brain and Google Deep Mind. Moreover, although time and energy are related, measuring energy consumption can offer a unique overview on top of measuring time, since there are cases where the execution time of an algorithm can be longer but the energy consumption lower (for example when using DVFS). Finally, although measuring energy consumption can be complicated, there are some solutions that can model the energy consumption of different algorithms [8]. In Section 6 we presented an approach to measure the energy consumption at the function level of any algorithm using the x86 simulator Sniper. The study concludes with an experiment where we show the energy consumption of an online decision tree under three different datasets.

The planned future work is to investigate the key operations that can reduce the overall energy consumption of different algorithms, together with a generic understanding of the energy complexity of decision trees.

Acknowledgments

This work is part of the research project Scalable resource-efficient systems for big data analytics funded by the Knowledge Foundation (grant: 20140032) in Sweden.

References

- [1] M. Naghavi, H. Wang, R. Lozano, A. Davis, X. Liang, M. Zhou, S. E. V. Vollset, A. Abbasoglu Ozgoren, R. E. Norman, T. Vos, *et al.*, “Global, regional, and national agesex specific all-cause and cause-specific mortality for 240 causes of death: a systematic analysis for the global burden of disease study 2013,” *The Lancet*, vol. 385, no. 9963, pp. 117–171, 2015.
- [2] T. Johann, M. Dick, S. Naumann, and E. Kern, “How to measure energy-efficiency of software: Metrics and measurement results,” in *Proceedings of the First International Workshop on Green and Sustainable Software*, pp. 51–54, IEEE Press, 2012.
- [3] M. Dubois, M. Annavaram, and P. Stenström, *Parallel computer organization and design*. Cambridge University Press, 2012.

- [4] C. Reams, *Modelling energy efficiency for computation*. PhD thesis, University of Cambridge, 2012.
- [5] A. Freire, C. Macdonald, N. Tonello, I. Ounis, and F. Casheda, "A self-adapting latency/power tradeoff model for replicated search engines," in *7th ACM international conference on Web search and data mining*, pp. 13–22, 2014.
- [6] A. Hooper, "Green computing," *Communication of the ACM*, vol. 51, no. 10, pp. 11–13, 2008.
- [7] S. Murugesan, "Harnessing green it: Principles and practices," *IT professional*, vol. 10, no. 1, pp. 24–33, 2008.
- [8] V. Sze, Y.-H. Chen, T.-J. Yang, and J. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *arXiv preprint arXiv:1703.09039*, 2017.
- [9] T.-J. Yang, Y.-H. Chen, and V. Sze, "Designing energy-efficient convolutional neural networks using energy-aware pruning," *arXiv preprint arXiv:1611.05128*, 2016.
- [10] E. García-Martín, N. Lavesson, and H. Grahn, "Energy efficiency analysis of the Very Fast Decision Tree algorithm," in *Trends in Social Network Analysis - Information Propagation, User Behavior Modelling, Forecasting, and Vulnerability Assessment. (To appear in April 2017)* (R. Missaoui, T. Abdesslem, and M. Latapy, eds.), 2016.
- [11] A. Noureddine, R. Rouvoy, and L. Seinturier, "Monitoring energy hotspots in software," *Automated Software Engineering*, vol. 22, no. 3, pp. 291–332, 2015.
- [12] E. García-Martín, N. Lavesson, and H. Grahn, "Identification of energy hotspots: A case study of the very fast decision tree," in *Green, Pervasive, and Cloud Computing. GPC 2017, LNCS 10232 (To appear in May 2017)* (M. A. et al, ed.), pp. 1–15, 2017.
- [13] T. E. Carlson, W. Heirman, S. Eyerman, I. Hur, and L. Eeckhout, "An evaluation of high-level mechanistic core models," *ACM Transactions on Architecture and Code Optimization (TACO)*, 2014.
- [14] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proceedings of the 6th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 71–80, 2000.
- [15] J. Gama, R. Rocha, and P. Medas, "Accurate decision trees for mining high-speed data streams," in *Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 523–528, ACM, 2003.
- [16] R. B. Kirkby, *Improving hoeffding trees*. PhD thesis, The University of Waikato, 2007.
- [17] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: Massive online analysis," *The Journal of Machine Learning Research*, vol. 11, pp. 1601–1604, 2010.