Optimizing emissions for machine learning training

Sachini Piyoni Ekanayake
Department of Electrical and Computer Engineering
University at Albany,
Albany, NY, USA
sekanayake@albany.edu

Tapan Shah
Machine Learning Lab
GE Research
San Ramon, CA, USA
tapan.shah@ge.com

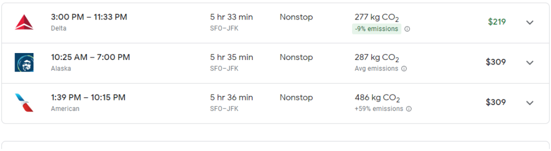
Scott Evans
Machine Learning Lab
GE Research
Niskayuna, NY, USA
evans@ge.com

I. Introduction

Modern machine learning models consume massive amounts of energy. In a widely cited paper [9], the authors compare the estimated CO2 emissions from training common NLP models like BERT [5], GPT-2 [1], ELMO [8] and transformers [10]. Similarly, in [7], the authors compare other models like Meena [2] and GPT-3 [4]. For example, GPT-3 training consumes around 550 metric tons of carbon. Together, data center use including machine learning model creation is projected to grow exponentially in the coming years [6]. As machine learning becomes used by more and more organizations for their business processes, it is imperative new paradigms for efficient model training and inferencing are developed. Our research project is motivated from a recent flight search which showed multiple flight options along with their estimated carbon emissions (Figure 1) . It is left upon the discretion of the traveller to pick a suitable flight based on CO2 emissions, comfort and convenience. We extrapolate the same for training a batch of ML jobs and create a scheduler which can appropriately allocate jobs to different datacenters at different times. In addition, based on the data characteristics and ML task (classification, regression etc.), we can also recommend the appropriate model to be used.

II. Problem Description

To illustrate the concept, we describe a simple problem below. Detailed notations, assumptions, constraints, optimization program formulation and the implementation are described in the Appendix. We would like to highlight that key objective of the project is to highlight the concept and hence some of the assumptions might not be very practical.



3:00 PM – 11:33 PM
Delta
5 hr 33 min
SFO–JFK
Nonstop
277 kg CO$_2$
-9% emissions
$219

10:25 AM – 7:00 PM
Alaska
5 hr 35 min
SFO–JFK
Nonstop
287 kg CO$_2$
Avg emissions
$309

1:39 PM – 10:15 PM
American
5 hr 36 min
SFO–JFK
Nonstop
486 kg CO$_2$
+59% emissions
$309

Fig. 1: A motivating example of how a flight research also shows estimation carbon emissions. These is factored in when the traveller makes which flight/route is chosen.

Consider a set of customers C$\triangleq$ $\{1,\ldots,J\}$ with J number of ML jobs and they are corresponding to specific machine learning tasks (e.g., classification, regression) and specific requirements e.g., required accuracy $\eta_j^{'}$ (*output quality*). Let a task $j, j = 1,\ldots,J$ arrive at time t_a^j. We assume that there exists a relevant dataset D_j as an input along with a task j. Thus, each j is characterized by:

1) Machine Learning algorithm (e.g., Support Vector Machine (SVM), Neural Network (NN))
   - Algorithm specifications and/or Quantifying parameters like hyperparameters (learning rate) and structural parameters (eg. number of hidden layers)
2) Quality specifications (e.g.,(expected) accuracy $\eta_j^{'}$ as a lower bound)
3) Dataset properties like sixe, preprocessing and feature selection
4) Energy requirement $E(j)$
5) Computation time $T(j)$

For simplicity, we assume that, among the possible set of algorithms that satisfy the given quality requirements, we run the algorithm that has the *best figure of merits*[1] i.e., max $\frac{\eta_j}{E(j)}$ to do the job j. To facilitate more practically realizable setting, and to incorporate importance of execution time, we consider that machine learning tasks have a priority i.e., HIGH, MEDIUM, LOW that affect the starting time. Consider that there exists H number of high performance computers (HPC) $\mathcal{H} \triangleq \{h|1,\ldots,H\}$ across L number of HPC locations (centers) $\mathcal{L} \triangleq \{l|1,\ldots,L\}$ where $L \leq H$. Next, we define location cluster $S_l \triangleq \{h|f_l(h) = l\}$ where $f_l(h)$ is a mapping function that takes set of HPCs h as input and assigned them to

[1] Figure of merit is defined as $\frac{\text{accuracy}}{\text{required energy}}$. This is a big assumption we make in our project. Typically, for a new dataset, we will not have an estimate of the figure of merit. We propose to use concepts of meta-learning to get estimate of these value from previously trained models.

particular location l. We consider there exists maximum of available power $P_l(t)^{\max}$ for a specific location l and is defined as $P_l(t)^{\max} \triangleq P_{l,c=0}(t)^{\max} + P_{l,c=1}(t)^{\max}$. Here, $P_{l,c=0}(t)^{\max}$ is the maximum available power from energy sources where carbon emission is 0 ($c = 0$) and $P_{l,c=1}(t)^{\max}$ is the maximum available power from carbon emitting ($c = 1$) power sources. Available power is discretized across a Z time horizon of $T = \{t_1, \dots, t_Z\}$. The total consumed power of a location l at time t is defined as $P_l(t)$ such that $0 \le P_l(t) \le P_l(t)^{\max}$. We assume that length of a $t_z, z = 1, \dots, Z$ is time block of Δt (e.g., 1hour). For instance, if we consider a whole day with $\Delta t = 1h$, $T = \{1, \dots, 24\}$ where $Z = 24$. The goal is to obtain optimum scheduling such that, task $j \in C$ can be assigned to HPC $h \in H$, and describe the distribution of energy $E(j)$ across h and t to minimize total carbon emitting energy consumption.

III. Example

Consider an example with 3 machine learning tasks and 3 HPC locations. The settings are described in Table I. We compare the optimal allocation and compare it with a random and a greedy allocation.

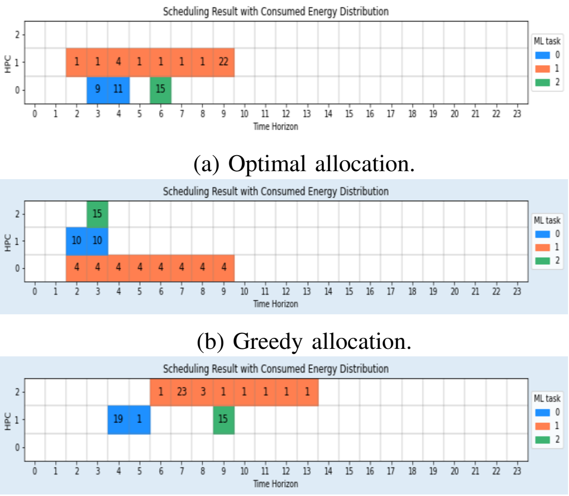
ML Task j	0	1	2
Expected Energy Requirement $E(j)$	20	32	15
Arrival Time t_a^j	2	2	3
Computation Time $T(j)$	2	8	1
Start Time Margin δ_t^j (hours)	2	4	6

TABLE I: An example scenario to illustrate the scheduling problem.

In Figure 2, we compare the optimal allocation with the greedy and random allocations. The number in each colored box indicates amount of energy to be consumed in that time stamp. In optimal allocation, this is so chosen so as to minimize carbon emission in that timestamp. Another point to note is that for task 2, the training starts much later, using the start time margin to minimize carbon emission. On an average, across multiple settings, we observed 10 times lesser carbon emission as compared to random allocation and 4 times lower as compared to greedy allocation.

IV. Conclusions

Due to space limitations, we only describe a simple setting to highlight how an optimal scheduling can greatly decrease the carbon emissions. There are 2 key assumptions made to solve the optimization problem: 1) The total energy and time requirement for an ML job is known apriori and 2) the power profile at any HPC location is known apriori. These are not unreasonable assumptions. Meta-learning is known to give an estimate for the first problem while modern power forecasting algorithms [3] give an answer to the second. However, optimization problem thus needs to be updated to incorporate the uncertainty in the estimates.



(a) Optimal allocation.

(b) Greedy allocation.

(c) Random Allocation.

Fig. 2: Spatio-temporal allocation of different ML tasks across different HPCs.

References

[1] OpenAI GPT2.

[2] Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. Towards a Human-like Open-Domain Chatbot. *arXiv:2001.09977 [cs, stat]*, February 2020. arXiv: 2001.09977.

[3] Adil Ahmed and Muhammad Khalid. A review on the selected applications of forecasting models in renewable power systems. *Renewable and Sustainable Energy Reviews*, 100:9–21, 2019.

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. *arXiv:2005.14165 [cs]*, July 2020. arXiv: 2005.14165.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, May 2019. arXiv: 1810.04805.

[6] Martijn Koot and Fons Wijnhoven. Usage impact on data center electricity needs: A system dynamic forecasting model. *Applied Energy*, 291:116798, June 2021.

[7] David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon Emissions and Large Neural Network Training. page 22.

[8] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv:1802.05365 [cs]*, March 2018. arXiv: 1802.05365.

[9] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and Policy Considerations for Deep Learning in NLP. *arXiv:1906.02243 [cs]*, June 2019. arXiv: 1906.02243.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv:1706.03762 [cs]*, December 2017. arXiv: 1706.03762.