AI and Environmental Impact Project

Akedjou Achraff Adjileye
Antoine Olivier

27 March 2024

Abstract

As deep learning technologies propel artificial intelligence forward, growing concerns about their environmental impacts, particularly the carbon footprint from training large-scale neural networks, have emerged. This study reviews the paper titled "The Energy and Carbon Footprint of Training End-to-End Speech Recognizers" 1 in which Parcollet et al. seeks to develop a comprehensive framework and methodology for accurately estimating emissions during the training of such models. Their findings highlight the urgent need for a holistic evaluation method that encompasses both performance metrics and energy efficiency. Adopting this approach is vital for the sustainable development and deployment of deep learning models, ensuring they align with environmental conservation efforts.

1 Introduction

This report aims to provide a clear and concise explanation of the method for estimating CO_2 emissions during the training of deep learning models, proposed in 1. We will first expose the method (section 2.1), then we will illustrate its application through concrete examples one from the paper (section 2.2) and additional experiments conducted by us (section 3). This analysis will facilitate a better understanding of the environmental impact of training artificial intelligence models and enable consideration of measures to reduce it.

2 Contribution of the paper

2.1 Framework

2.1.1 CPU and GPU energy consumption

Estimating consumption based on hardware power specifications does not provide a sufficiently accurate estimate, as the hardware is typically only partially utilized during training. To address this issue, it is recommended to use tools such as **Carbon Tracker** 5 or **codecarbon** 4 to obtain real-time device consumption.

2.1.2 Power Usage Effectiveness

When training of large models, energy consumption does not solely stem from the usage of CPUs and GPUs. A significant portion comes from cooling, which is why the paper recommends estimating the efficiency of energy usage (PUE: Power Usage Effectiveness), the ratio of total energy consumed P_{facility} to the energy used by computing units (CPUs and GPUs) P_{compute}, and using it to calculate the total consumption of the facility during training .

$$PUE = \frac{P_{\text{facility}}}{P_{\text{compute}}}$$

For reference, the average PUE in the academic field is estimated to be around 1.55.

2.1.3 Energy comsumption estimation

Thus, from the energy consumption of CPUs and GPUs $P_{\text{CPUs/GPUs}}$, we can calculate the total energy consumed as follows:

$$e_{total} = PUE \times P_{CPUs/GPUs}.$$

2.1.4 CO_2 Emission

To estimate the quantity of emitted CO_2, one can calculate the conversion by multiplying the estimated total energy consumption by the conversion rate c_{rate} in CO_2 (gCO_2/kWh):

$$T_{carbon} = c_{rate} \times e_{total}.$$

This rate varies by country depending on the nature of electricity production, such as nuclear or fossil sources. These rates are available in real-time on Electricity Map.

2.2 Experiments and Discussion

2.3 Experiments

Authors of applied their framework to estimate the CO_2 emissions generated during the training of various large models for automatic speech recognition (ASR). Models architectures include CTC-Attention based Transformers 8, CTC-Attention based CRDNN 9, and RNN-Transducers 10. These models were trained using different GPUs (Tesla V100 as high-end and RTX 2020 Ti as mid-tier), and on different datasets (Common Voice 11 and LibriSpeech 12). They trained each model for 3 epochs to assess the total training cost in terms of CO_2 emissions.1, calculated for different countries (France and Australia). The evaluation metric is the word error rate (WER).

	CommonVoice					LibriSpeech				
Tesla V100	**kWh per epoch**	**Epochs**	**CO_2 France (kg)**	**CO_2 Au. (kg)**	**WER %**	**kWh per epoch**	**Epochs**	**CO_2 France (kg)**	**CO_2 Au. (kg)**	**WER %**
CRDNN (1,1)	2.11	25	2.77	34.66	17.70	3.78	25	4.92	62.04	2.90
Transformer (1,1)	0.92	40	1.92	24.20	20.57	1.49	121	9.38	118.4	2.55
RNN-T (1,3)	2.00	30	3.12	39.35	20.18	6.58	30	10.26	129.5	5.23
RTX 2080 Ti										
CRDNN (3,3)	5.28	25	6.87	86.63	17.70	5.40	25	7.01	88.45	2.90
Transformer (3,8)	2.98	40	6.19	78.13	20.57	1.72	121	10.87	137.1	2.55
RNN-T (3,6)	4.37	30	6.83	86.16	20.18	8.37	30	13.06	164.7	5.23

Figure 1: Extracted from 1: CO_2 and energy consumption estimates for popular E2E ASR models trained with computational resources located in France or in Australia (Au.). The Word Error Rates (WER) on CommonVoice (CV) FR and LibriSpeech (LS) are obtained on the "test" and "test-clean" sets of CV and LS respectively. The (x, y) given with model name indicates the number of GPUs used for (CV,LS).

Figure (2) shows the CO_2 emissions of different trainings against the WER.

2.4 Discussion

Here are some of the conclusions of 1:

Geographic Location Impact: The amount of CO2 emitted during training varies significantly depending on the geographic location, with a factor of 10 difference observed between France and Australia, for example.

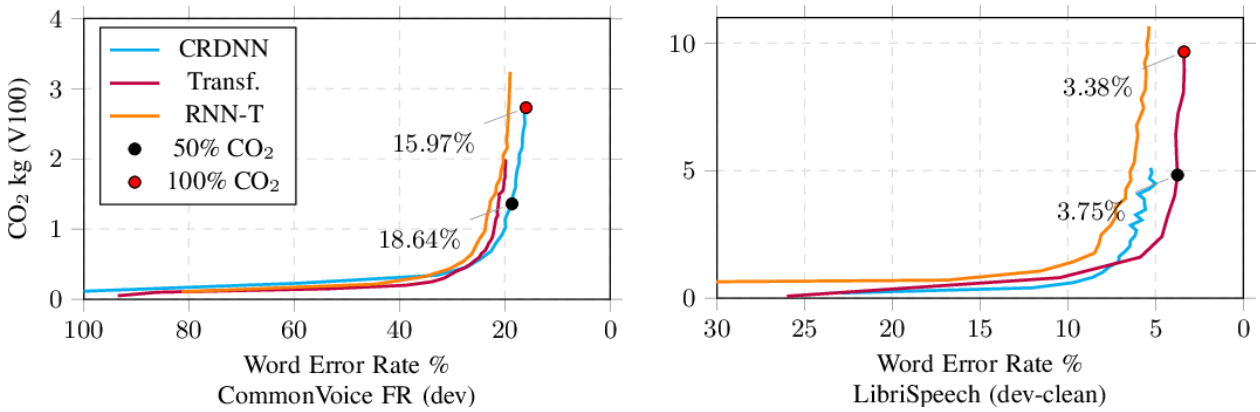

Figure 2: Extracted from 1: CO_2 emitted in kg (in France) by different E2E ASR models with respect to the word error rate (WER) on the dev sets of LibriSpeech and CommonVoice. The curves exhibit an exponential trend as most of the training time is devoted to slightly reduce the WER. The black and red dots indicates the WER obtained with 50% and 100% of the emitted CO2. On LibriSpeech, 50% of the carbon emissions have been dedicated to reach SOTA results with an improvement of 0.37%.

GPU Type Significance: Recent GPUs play a crucial role due to their efficiency, with a 2.5-fold difference observed between experiments conducted with RTX 2080 Ti and Tesla V100 GPUs.

Hyperparameter Search Impact: Optimizing hyperparameters contributes significantly to CO_2 emissions. Conducting multiple experiments to find the ideal configuration can result in considerable emissions. This underscores the necessity for improved tuning strategies.

Impact of Model Architecture: Although different architectures have varying energy consumption per epoch, they tend to converge to similar CO_2 emissions due to differences in convergence time.

Unreasonable cost of state-of-the-art performance: Figure 2 shows the significant cost associated with achieving state-of-the-art performance. The analysis shows that as models aim for lower word error rates, the corresponding CO_2 emissions increase exponentially.

3 Experiments On A New Task

Inspired by the experiments conducted in 1, which demonstrates an exponential increase in CO_2 emissions as a function of WER, we aimed to verify this result for the training of a **language model**. Due to our limited resources (one T4 GPU provided for free by kaggle for a limited time), we conducted our experiments on a tiny model with 1 million parameters. The task involves automatically classifying high school mathematics exam exercises into four categories based on themes: functions, numerical sequences, probabilities, and geometry in space. We created a dataset of 77 exercises, with 26, 15, 18, 18 for each of the mentioned categories, respectively. The dataset was created from LaTeX code of complete French high school mathematics exam (BAC) papers from recent years; we manually split each exam to have the files for the LaTeX codes of the exercises, making sure to add the headers (packages, format) for each exercise file. The 77 exercises were extracted from 19 complete exam papers, and the whole preparation process lasted between **3 and 4 hours**.

3.1 Pretraining: Masked Language Modeling (MLM)

Pre-training language models on masked language modeling **MLM** tasks before fine-tuning them on more complex tasks such as document classification is well-known as a crucial step in natural language processing (NLP) workflows. Masked language modeling involves training a model to predict masked words within a given text, which helps the model learn contextual representations and semantic relationships between words. This pre-training phase is essential as it enables the model to capture a broad understanding of language patterns and structures across diverse datasets. Therefore, we considered MLM as our pretraining task. Our goal at this stage was to ensure that the model learns, so we did not perform a validation split and used all the available texts for training.

3.1.1 Technical details

Data tokenization: Since our documents are LaTeX codes, we created our own tokenizer using **Byte-Pair Encoding** 2, obtaining a vocabulary of **3824** tokens. Then, each document is split into chunks and tokenized so that the token vector size is **512**, the input size of our models. We use padding to fill the chunks if necessary. This generated **148** segments of size 512; we randomly mask **15%** of the tokens at each forward pass of the training.

Model architecture: The model used is an instance of the **RoBERTaForMaskedLM** class 3 in the **HuggingFace Transformers** library, with hyperparameters reported in Table 1.

Hyperparameters	Values
num_attention_heads	2
hidden_size	128
intermediate_size	512
num_hidden_layers	2

Table 1: Our RoBERTa model hyperparameters

Training parameters: We trained the model for **25K** epochs with a batch size of **148** (all segments) at a learning rate of **1e-4**, using a linear scheduler with **0.01** warmup ratio, on a machine with 1 **Tesla T4 GPU** and 4 **Intel(R) Xeon(R) CPUs** . During training, we tracked the energy consumption with the **EmmissionsTracker** of the **CodeCarbon** 4 library. The training lasted **2 hours 17 minutes**. The loss is the CrossEntropy.

Figure 3 shows the opposite of the loss (the higher, the better) as a function of energy consumption during the training process.

We observe an exponential increase of the energy consumption against the loss, which highlights the excessive energy cost that small performance gains can represent. In the following, we will show that, in

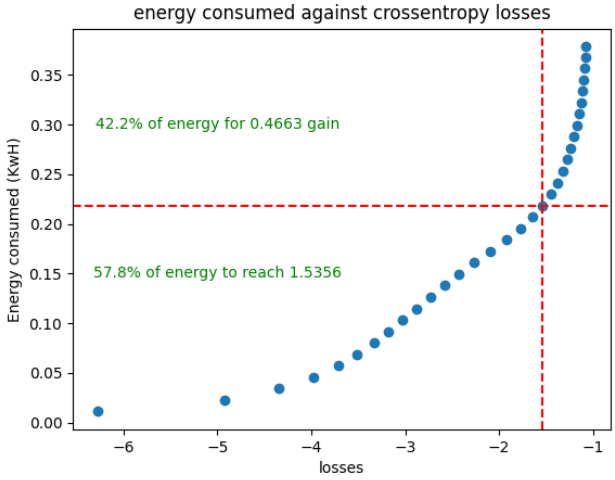

Figure 3: Energy consumption of the MLM pretraining, the figure shows an exponential evolution in relation to the crossentropy. More than **42.2%** of the energy consumption is used to gain **0.4663** in loss, confirming the excessive energy cost of state-of-the-art (SOTA) performances highlighted in 1

addition to being very costly in terms of energy, this costly performance gain is ridiculous as it has a **negligible** influence on the model's performance after being finetuned on the classification task.

3.2 Finetuning: Math Exercises Classification

After the pretraining done, we finetuned the model on exercise classification. Each exercise was tokenized in the same way as the pretraining chunks. We split the exercises into a training part (50%) and a validation part (50%), ensuring to stratify by labels to maintain the same label proportions in each split. Three versions of the model were finetuned on the training split and evaluated on the validation split: a checkpoint at 15K epochs of pretraining (corresponding to the lower-left corner zone in Figure 3), the checkpoint at the end the 25K epochs corresponding to the entire pretraining, and a version trained from scratch. Our goal in finetuning the 15K epochs checkpoint was to measure the effect of the energy costly performance gain during the 10K additional pretraining epochs, on the model's performance in finetuning. However, training from scratch confirms the importance of the pretraining phase for such model. To ensure fairness, we trained all three models with the same learning rate of **1e-5** using a linear scheduler with 0.01 warmup ratio, a batch size of **38** (all training samples) for **1.5K** epochs. Figure 4 shows the accuracy curves (on the validation split) during training.

The first observation is that both finetuned models (15K and 25K) achieve the same accuracy of **0.923** at the end, which is significant as it means that the additional 10K epochs of pretraining, representing 42.2% of the total energy consumption, do not make a difference in the classification performance. Although the orange curve seems to converge faster than the blue curve, it likely does not justify such additional energy consumption during pretraining. This result definitively confirms the excessive and ridiculous cost of the SOTA highlighted in 1 and raises questions about how models should be benchmarked: **should energy costs and consequently carbon footprint not be indicators to systematically take into account?**

In other hand, the trainings from scratch confirm the importance of the pretraining phase. The green curve in Figure 4 shows the model's difficulties to perform well without pretraining, even when allowed to train for a longer period (red curve).

These results show the excessive cost of obtaining SOTA performance during the pretraining phase, its uselessness in a context where the ultimate goal is to finetune the model on another task of interest, but also highlight the importance of the pretraining phase. The question that naturally arises then is **how to determine the right time to stop the pretraining process to ensure SOTA performance in finetuning and**

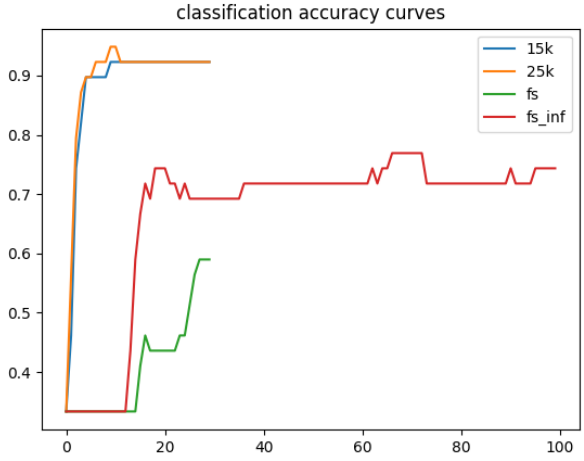

Figure 4: Classification accuracy curves during the finetuning phase: orange/blue curves are for the 25K/15K epochs pretrained checkpoint, green/red curves are for the model trained from scratch (fs) on classification for 1.5K/5K epochs. Training the model longer for 5K epochs (fs_inf) shows that the model still does not converge to top performance even when trained for a long time.

avoid unnecessary energy expenses.

The figure 5 displays the energy consumption of the finetuning process of the 15K and 25K epochs checkpoints against the accuracy scores.

Again, we can see the excessive energy cost of aiming for SOTA performance in the specific finetuning task, with more than 57% of energy consumption not providing any performance surplus. This is another proof of the fact that one should definitely brings the energy costs in the loop when training these neural models.

3.2.1 About Carbon Emissions

Table 2 presents the carbon emissions and information about servers localization of our computations; provided by the CodeCarbon's **EmissionsTracker** 4 .

Models	**ft 25k ckp**	**fs_inf**	**pretr mlm_25k**
Duration	221.80	731.86	8216.97
Energy Consumed	0.0080	0.0272	0.3784
PUE	1.0	1.0	1.0
Emissions (KgCO_2eq)	0.0036	0.0123	0.0525
Country Name	United States	United States	United States
Region	Iowa	Iowa	Oregon

Table 2: Emissions of our computations: "ft 25k ckp" refers to the finetuning of the checkpoint pretrained for 25k epochs, "pretr mlm_25k" denotes the pretraining for 25k epochs on MLM, and "fs_inf" indicates the long (5K epochs) training from scratch on exercises classification.

4 Limitations and Conclusion

In this study, we reviewed the paper 1 on estimating the energy and carbon footprint of end-to-end training for ASR system. The paper highlights the excessive energy cost and carbon footprint associated with achieving

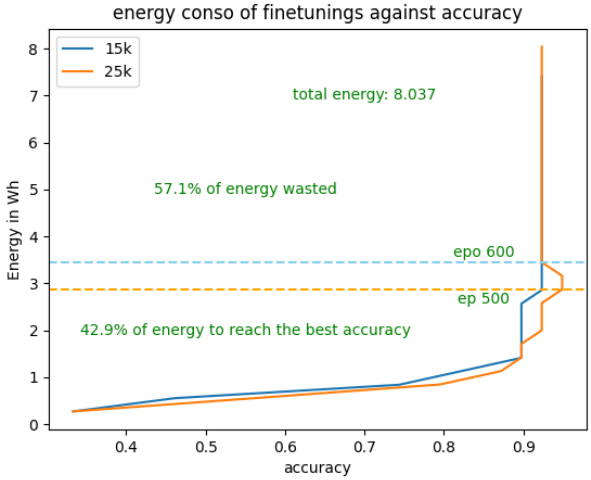

Figure 5: Energy consumed by the two finetuned models: 15K epochs of pretraining and 25K epochs. Energy units are in Wh not KWh; indeed, finetuning is much faster and therefore consumes much less energy than pretraining.

state-of-the-art (SOTA) performances for these models.

We tested these findings on the end-to-end training of a RoBERTa-based model for mathematic exercises classification and arrived at similar conclusions. Specifically, **42.2%** of the energy consumed during the pretraining of our model was found to be unnecessary for achieving good performance on the final task. Similarly, **57.1%** of the energy consumed during the model's finetuning on the task of interest was deemed unnecessary. These proportions are extremely high, and such experiments conducted at a small scale (model with 1 million parameters) send strong signals about the need to consider energy costs during the training of such systems to achieve SOTA performance, especially in the era of **foundation models** 7. Moreover, most of these systems are intended for large-scale application use, and often the minor performance gain obtained at an unreasonable energy cost often goes unnoticed by users of these systems.

However, our training experiments with different models did not include hyperparameters optimization; we simply set appropriate values to achieve reasonable performance to evaluate the energy costs. This is why, for example, we did not attempt to adjust the learning parameters of the model pretrained for 25K epochs when it reached an accuracy of **0.9487** after 500 epochs (fig 5). Determining whether this rapid achievement is due to longer pretraining or aleatoric uncertainty remains to be investigated, as does whether finetuning the model pretrained for 15k epochs could have also reached better accuracy with better hyperparameters tuning. In any case, these experiments would still consume energy and would once again confirm the excessive cost of achieving SOTA performance and the need to include this aspect in the loop when benchmarking neural models.

5 References

1. Titouan Parcollet, Mirco Ravanelli. The Energy and Carbon Footprint of Training End-to-End Speech Recognizers. 2021. hal-03190119

2. Byte Pair Encoding

3. RoBERTa HuggingFace

4. CodeCarbon, Track and reduce CO_2 emissions from your computing.

5. Carbon Tracker

6. L. F. W. Anthony, B. Kanding, and R. Selvan, “Carbontracker: Tracking and predicting the carbon footprint of training deep learning models,” ICML Workshop on Challenges in Deploying and monitoring Machine Learning Systems, July 2020, arXiv:2007.03051.

7. Foundation Model

8. S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang et al., “A comparative study on transformer vs rnn in speech applications,” in 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2019, pp. 449–456.

9. M. Ravanelli, T. Parcollet, A. Rouhe, P. Plantinga, E. Rastorgueva, L. Lugosh, N. Dawalatabad, C. Ju-Chieh, A. Heba, F. Grondin, W. Aris, C.-F. Liao, S. Cornell, S.-L. Yeh, H. Na, Y. Gao, S.- W. Fu, C. Subakan, R. De Mori, and Y. Bengio, “Speechbrain,https://github.com/speechbrain/speechbrain, 2021.”

10. A. Graves, “Sequence transduction with recurrent neural networks,” arXiv preprint arXiv:1211.3711, 2012

11. R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. Tyers, and G. Weber, “Common voice: A massively-multilingual speech corpus,” in Proceedings of The 12th Language Resources and Evaluation Conference, 2020, pp. 4218–4222.

12. V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2015, pp. 5206–5210.