Performance above all ? energy consumption vs. performance for machine listening, a study on dcase task 4 baseline

Romain Serizel, Samuele Cornell, Nicolas Turpault

PERFORMANCE ABOVE ALL ? ENERGY CONSUMPTION VS. PERFORMANCE FOR MACHINE LISTENING, A STUDY ON DCASE TASK 4 BASELINE

Romain Serizel[1]*, Samuele Cornell*[2]*, Nicolas Turpault*[3]

[1] Universite de Lorraine, CNRS, Inria, Loria, Nancy, France
[2]Department of Information Engineering, Universita Politecnica delle Marche, Italy
[3]Inria centre at Rennes University, Rennes, France

ABSTRACT

In machine listening there is a tendency to resort to models with a growing number of parameters raising thus concerns about the practical viability of these due to their energy consumption. Reporting energy consumption of the models could be a first step to raise awareness on this matter. Yet, estimating the energy consumption across different conditions (hyper-parameters, GPU types etc.) poses some challenges in terms of biases and fairness of the comparison between different models and works. In this paper we perform an extensive study using the DCASE task 4 baseline system and monitor energy consumption and training time for different GPU types and batch sizes. The goal is to identify which aspects can have an impact on the estimation of the energy consumption and should be normalized for a fair comparison across systems. Additionally, we propose an analysis of the relationship between the energy consumption and the sound event detection performance that calls into question our current way to evaluate systems.

Index Terms— sound event detection, machine listening, energy consumption, efficiency, carbon footprint

1. INTRODUCTION

Deep learning has enabled significant progress in the field of machine listening, and is now the de-facto standard approach for tackling problems and applications such as sound event detection (SED) and recognition (SER). However as argued by many works [1–4] some deep learning recent trends, both in industry and academia, raise fundamental concerns about the environmental impact and energy consumption of deep learning approaches. For example in [2], the authors point out that the large majority of the current research is focused on absolute performance rather than model efficiency. The downside of this is that, in applications such as natural language processing (NLP), the computing requirements for training state-of-the-art models are increasingly prohibitive and the actual costs due to hardware and to energy consumption can amount to tens of thousands of dollars. This also often translates in expensive inference too, which can be even more problematic: when the model is deployed in a real-time application, its impact over time will largely surpass the training stage one. Moreover, if it must be deployed server-side because it is too demanding for edge-devices, data transmission will add up to the environmental/cost impact of the model. On the other hand, large-scale models are undoubtedly useful and there is a growing interest towards more efficient ways to tap their potential in a cost-effective and environmental-friendly manner. For example, by model compression and/or quantization techniques [5,6] or by leveraging knowledge distillation [7].

Recently in [4], Parcollet and Ravanelli, investigated the carbon footprint of training state-of-the-art medium size automatic speech recognition (ASR) models. It was pointed out how training a state-of-the-art ASR model on the popular LibriSpeech benchmark dataset [8] can exceed several times the carbon emissions of driving a car for 100 km. Crucially, in their study, increasing the model size and doubling the emissions only improved marginally the performance, raising thus questions about if such improvement is worth or not considering the financial and environmental costs.

1.1. Motivations and background

For the past few years DCASE task 4 has proposed to investigate SED. The particular use case is to train the SED system on an heterogeneous dataset composed of audio soundscapes with different level of annotations. The problem of evaluating systems under realistic setups in order to obtain insights on the systems (and just a ranking) has been central in the task organization [9–12].

Over the years there has been a tendency, in line to what is argued in [2], to resort to models with a growing number of

This work was made with the support of the French National Research Agency, in the framework of the project LEAUDS Learning to understand audio scenes (ANR-18-CE23-0020). Experiments presented in this paper were carried out using the Grid5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000).

parameters which are even sometimes combined with each other using ensemble approaches[1]. As said, this tendency raises concerns about the financial and environmental viability of these models. As DCASE task organizers we felt it was our duty to relay this concern to the participants and this is why in the recent DCASE 2022 task 4 we asked participant to report the energy consumption of their systems both at training and test time [13].

Reporting the energy consumption was introduced as an optional metric as its estimation on the participants side could raise some question in terms of biases (e.g. the hardware used) and fairness of the comparison between systems. These concerns were confirmed to some extent by the submissions we received this year. As such, one of the goals of this paper is to identify some potential sources of variability in the estimation of the energy consumption on the participants side. Here we analyze the energy consumed in the training phase of the DCASE 2022 task 4 baseline when the GPU type and batch size are changed, while at the same time monitoring the performance of the SED model. We are fully aware that the energy consumed at inference phase plays a large role in the overall footprint of a system. However, here we focus only on the training phase as a similar analysis on the test phase requires knowledge of the specific hardware used when deploying the SED algorithms (especially on edge-devices) which is beyond the focus of the current paper.

Our work is similar thus to [4] but, compared to this one, aside from focusing on SED and energy consumption instead of carbon oxide emissions, we explore also factors of variations such as GPU type and how hyper-parameters as batch size have also an impact.

Finally we hope that this work could give some insights and "best-practices" that allow to reach the best trade-off between performance and energy consumption for SED applications when an heterogeneous dataset is used.

2. EXPERIMENTAL SETUP

2.1. SED System

The DCASE 2022 task 4 baseline system is based on a convolutional recurrent neural network (CRNN) trained within a mean-teacher framework. This system is inspired by [14]. The CRNN is composed of a 7 layers CNN module followed by a 2-layers bi-directional gated recurrent unit (biGRU). Input features are 128 Log-Mel Filterbank Energies (LFBE) extracted with a 128 ms window and 16 ms stride. To leverage more effectively weakly, unlabeled and labeled data, attention pooling is employed to obtain clip-level predictions from frame-level predictions [14]. With respect to the 2020 baseline, during the years, small changes, such as MixUp [15], and some hyper-parameters improvements have been added to this simple but effective baseline system, based on top-ranked systems submitted every year from participants[2].

2.2. Dataset

The dataset used for the SED experiments is DESED[3], a dataset for SED in domestic environments composed of 10-sec audio clips that are recorded or synthesized [9, 16]. The recorded soundscapes are taken from AudioSet [17]. The synthetic soundscapes are generated using Scaper [18]. The foreground events are obtained from FSD50k [19, 20]. The background textures are obtained from the SINS dataset [21] and TUT scenes 2016 development dataset [22].

2.3. Evaluation Metric

Within DCASE task 4, systems performance is evaluated with the polyphonic sound event detection scores (PSDS) [23]. However, in this study we monitor the performance at training time each epoch and PSDS would be too costly in such scenario. Therefore, we instead use the intersection based F1-score as a performance metric. The decision threshold is set to 0.5, the detection tolerance criterion (DTC) and ground truth intersection criterion (GTC) are set to 0.7. This is considerably faster to compute as it relies on a single operating point (threshold value) but still alleviates the problem of event matching with collar based metrics [10, 23]. Additionally, this metric has been shown to correlate with PSDS scenario 1 [11]. This is because the operating point considered to compute the F1-score is part of the set of operating points included in PSDS computation and the DTC and GTC are set similarly as when computing PSDS for scenario 1.

The energy consumption at training time is estimated using the CodeCarbon toolkit [4], a software package that estimates the amount of energy consumed and carbon dioxide produced by the cloud or personal computing resources used to execute the code. The energy consumption is monitored for each epoch and aggregated over the whole training phase.

The baseline system is trained 3 times for 200 epochs for each combination of batch size and hardware, this allows us to report also standard deviation over these 3 runs.

2.4. Hardware

One of the goals of this study is to analyze the impact of the hardware used at training time (and in particular the GPU type) on the energy consumption. Compared to previous studies, here we consider a wider set of GPUs commonly employed in academia and industry alike, both newer and older: GTX 980, GTX 1080 Ti, RTX 2080 Ti, T4, A40 and A100. To obtain correct energy consumption figures, for each experiment, the GPU is always allocated to training only our model with no other task running in parallel.

[1] see also the DCASE task for results pages on https://dcase.community/

[2] Code is publicly available: github.com/DCASE-REPO/DESED_task

[3] https://project.inria.fr/desed/

[4] https://codecarbon.io/

Batch	Max F1	Time	Energy Total (kWh)	Energy Epoch(Wh)	Energy Minute(Wh)
4	49.1 ± 0.7	08:28 ± 0:45	1.27 ± 0.13	6.4 ± 0.7	2.6 ± 0.1
8	55.2 ± 0.8	06:39 ± 0:27	0.92 ± 0.08	4.6 ± 0.4	2.4 ± 0.1
16	**59.6 ± 0.4**	**06:20 ± 0:07**	0.98 ± 0.05	4.9 ± 0.3	2.7 ± 0.2
32	58.7 ± 0.4	**06:19 ± 0:07**	**0.83 ± 0.02**	**4.1 ± 0.1**	**2.2 ± 0.1**
GPU	Max F1	Time	Total (kWh)	Epoch(Wh)	Minute(Wh)
GTX 980	56.2 ± 0.4	11:38 ± 0:06	1.23 ± 0.03	6.2 ± 0.1	1.8 ± 0.1
GTX 1080 Ti	56.9 ± 0.7	09:10 ± 0:09	1.41 ± 0.02	7.0 ± 0.1	2.5 ± 0.1
RTX 2080 Ti	56.6 ± 0.6	04:31 ± 0:11	**0.59 ± 0.07**	**3.0 ± 0.4**	**2.2 ± 0.3**
T4	**56.1 ± 0.8**	06:28 ± 0:18	0.89 ± 0.25	4.5 ± 1.2	**2.3 ± 0.3**
A40	**56.4 ± 0.6**	05:12 ± 0:02	1.24 ± 0.05	6.2 ± 0.2	4.0 ± 0.2
A100	55.9 ± 0.4	**04:23 ± 0:06**	0.72 ± 0.01	3.6 ± 0.1	2.7 ± 0.1

Table 1: Systems performance, training time and energy consumption for different GPU types. Top panel: average over all the GPUs types. Bottom panel: average over all batch sizes.

2.5. Batch Size and Composition

Previous studies have shown empirically that composing a batch of $\frac{1}{4}$ weakly labelled recorded soundscapes, $\frac{1}{4}$ strongly labelled synthetic soundscapes and $\frac{1}{2}$ unlabelled recorded soundscape offers a good balance to train effectively a SED model on an heterogeneous dataset [24]. In this study we keep this batch composition and experiment with the following batch sizes: 4, 8, 16 and 32. We upper bounded the batch size to 32 in order to allow for running the experiments on all the GPU types without incurring in out-of-memory issues.

3. RESULTS AND DISCUSSIONS

We present our results and discuss them thereafter. As explained, we ran a total of 72 experiments: 3 runs for 4 different batch-sizes and 6 different types of GPUs.

3.1. Impact of Batch Size

The aim of this experiment is to study the impact of the batch size on the energy consumption at training time. The batch size could potentially have an impact on the computational load of the GPU but also on the overall training speed. In Table 1 upper panel, we present the performance averaged over 3 runs of 200 epochs and over all the 6 GPU types we used in our experiments. The F1 score we report is the best score achieved over the total 200 epoch. We also report the training time, the total energy consumption in kWh and the average energy consumption for each epoch and for each minute.

The overall training time and energy consumption varies significantly with the batch size. The general tendency is that both the training time and energy consumption decrease when increasing the batch size. This is also reflected on the energy consumed for one epoch. This is not surprising as the energy consumption remains stable over the epochs. If we consider the energy consumed in 1 minute, the difference when changing the batch size is barely significant. This indicates that the computational load of the GPU has a minor impact on the energy consumption. This aspect also explains the fact that the energy consumed over the 200 epoch is lower with the configurations that tend to train faster with larger batches while not consuming significantly more energy per time unit. Therefore, as the F1-score increases with the batch size (at least for the sizes presented here), using larger batch sizes is a reasonable choice when training SED systems. Note that these conclusions still holds when breaking the results per batch and per GPU[5].

3.2. Impact of Hardware

The aim of this experiment is to study the impact of the GPU used at training time on the energy consumption. In Table 1 bottom panel, we report the same metrics as previously except that they are averaged now over batch sizes and presented depending on the GPU used during training.

As expected, the best F1-score achieved remains quite stable across all GPU types. The energy consumption for the whole training phase varies significantly depending on the GPU that is used. Differently than what observed for the batch size, the energy consumed in 1 minutes also varies significantly with the GPU types. This indicates that variation in the overall consumption are directly related to the GPU type and not only to the training time. However, the training time can still have a large impact. For example, GTX 980 has the lowest energy consumption per minute but as the training is much longer than with other, newer GPUs, its overall consumption is amongst the largest.

This impact of the GPU used should be taken into account when comparing systems that are trained on different hardware. During DCASE 2022 task 4, to mitigate this problem, we proposed the participants to normalize the energy consumption of the submitted system by the energy consumed to train this very same baseline model on their hardware [11].

[5] See the detailed results

			Energy (kWh)				Extra cost
Batch	Max F1	95% Max F1	@F1 = 40%	@95% Max F1	@Max F1	200 epochs	95%→Max F1
4	49.1 ±0.7	46.6 ±0.7	0.17 ± 0.02	**0.24±0.05**	**0.38±0.06**	1.27 ± 0.13	**60%**
8	55.2 ±0.8	52.4 ±0.8	**0.14 ± 0.01**	0.27±0.05	0.56 ±0.15	0.92 ± 0.08	108%
16	**59.6 ±0.4**	**56.6 ±0.4**	0.16 ± 0.01	0.42±0.04	0.81 ±0.14	0.98 ± 0.05	100%
32	58.7 ±0.4	55.7 ±0.4	0.16 ± 0.01	0.33±0.03	0.60 ±0.08	**0.83 ± 0.02**	87%
GPU	Max F1	95% Max F1	@F1 = 40%	@95% Max F1	@Max F1	200 epochs	95%→Max F1
GTX 980	56.2 ± 0.4	53.3 ± 0.3	0.20 ± 0.01	0.42 ± 0.03	0.77 ± 0.17	1.23± 0.03	**80%**
GTX 1080 Ti	56.9 ± 0.7	**54.0 ± 0.5**	0.22 ± 0.01	0.51 ± 0.05	0.92 ± 0.07	1.41± 0.02	85%
RTX 2080 Ti	56.6 ± 0.6	**53.8 ± 0.3**	**0.09 ± 0.01**	0.26 ± 0.03	**0.37 ± 0.06**	**0.59± 0.07**	**80%**
T4	**56.1 ± 0.8**	53.3 ± 0.7	0.13 ± 0.04	0.26 ± 0.07	0.51 ± 0.15	0.89± 0.25	100%
A40	**56.4 ± 0.6**	53.6 ± 0.6	0.20 ± 0.01	0.45 ± 0.06	0.81 ± 0.13	1.24± 0.05	**80%**
A100	55.9 ± 0.4	53.1 ± 0.2	0.11 ± 0.01	**0.21 ± 0.02**	0.42 ± 0.02	0.72± 0.01	97%

Table 2: Systems performance and energy consumption at different convergence steps and for different GPU types. Top panel: average over all the GPU types. Bottom panel: average over all the batch sizes.

3.3. Energy consumption and SED Performance

In the previous experiments we used a fixed number of epoch of 200. Whereas the energy consumption remains stable over the whole training phase, this is not the case for the F1-score and there seems to be no linear relationship between the energy consumed and the F1-score performance. Therefore, in this experiment, we compare the energy consumed to achieve a fixed threshold F1-score of 40%, to achieve the maximum F1-score and to obtain 95% of this maximum F1-score. In Table 2 upper panel we report the F1-score and the energy consumption depending on the batch size (and averaged over GPU types). In Table 2 bottom panel, instead we show the F1-score and the energy consumption depending on the GPU types (and averaged over the batch sizes).

In the early stage of the training process, the batch size does not have a large impact on the energy consumed. The energy consumed to reach an F1 score of 40% is pretty stable across batch sizes. This does not verify for GPU types where the energy consumed to attain the 40% threshold can vary significantly. When considering the energy consumed to achieve the maximum F1-score, both the batch sizes and the GPU type do have an impact. For small batch sizes, the energy consumed is low but the F1-score is low too whereas for larger batch sizes the F1-score gets better but the energy consumption also increases. Note that at some point increasing the batch size maintain the F1-score performance while reducing the energy consumed. This also correlates with the fact that for small batch sizes the maximum performance is reached early in the training phase (hence the low total energy consumed due to the low number of epochs), while for larger batch sizes the maximum F1-score is reached later in the training process (reaching the maximum F1-score costs about 80% of the total training energy budget).

The observations above raise the question of the energy budget we allocate to gain the final few point in terms of F1-score as also noted in [4]. Regardless of the batch size or the GPU type, it costs 60% to more than 100% more to gain the last 5% relative in term of F1-score. This additional cost does not depend on the GPU type. It is however smaller for small batch size which converge quicker to the maximum performance. Such results suggest that comparing systems on the sole performance criterion is inadequate or at least does not provide the full picture (this is particularly true in challenges). It is important to asses the impact of these last few F1-score point on the systems usability (or if they actually matter for user experience) to ensure that the energy (and financial) budget spent for these minor improvements is actually worth it. A first step toward this is to systematically include the energy consumption budget in our performance eports [13, 25].

Finally, we stress that energy consumption can be reduced by using early stopping. Stopping the training at the best epoch could save about 40% of energy in average (up to 70% for small batch sizes) whereas stopping at 95% of the maximum F1-score could save about 65% in average (up to 80% for small batch sizes). While this latter solution is not realistic in practice, one easy alternative would be to stop training when the progress between epochs slows down too much.

4. CONCLUSION

We conducted an in depth study using the DCASE task 4 baseline for analyzing the relationships between energy consumption, training time, GPU type, batch size and performance when training a SED system. Our experiments show that the batch size impact on energy consumption is solely related to a reduction in training time. The GPU type used at training on the other hand has a large impact on the energy consumption. This aspect should be taken into account to avoid any biases when comparing systems trained on different GPUs. Finally, it was shown that the final training steps required to achieve the last few points in terms of SED performance cause a large increase in terms of energy consumption (between 60% to 100% more). This latter aspect calls into question our current way evaluate SED systems (machine listening systems in general) focusing only on performance.

5. REFERENCES

[1] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres, "Quantifying the carbon emissions of machine learning," *arXiv preprint arXiv:1910.09700*, 2019.

[2] Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni, "Green ai," *Proc. of ACM*, vol. 63, no. 12, pp. 54–63, 2020.

[3] Lasse F Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan, "Carbontracker: Tracking and predicting the carbon footprint of training deep learning models," in *ICML Workshop on Challenges in Deploying and monitoring Machine Learning Systems*, 2020.

[4] Titouan Parcollet and Mirco Ravanelli, "The energy and carbon footprint of training end-to-end speech recognizers," 2021.

[5] Pierre Stock, Angela Fan, Benjamin Graham, Edouard Grave, Rémi Gribonval, Herve Jegou, and Armand Joulin, "Training with quantization noise for extreme model compression," in *ICML*, 2021.

[6] Bowen Shi, Ming Sun, Chieh-Chi Kao, Viktor Rozgic, Spyros Matsoukas, and Chao Wang, "Compression of acoustic event detection models with low-rank matrix factorization and quantization training," *arXiv preprint arXiv:1905.00855*, 2019.

[7] Gianmarco Cerutti, Rahul Prasad, Alessio Brutti, and Elisabetta Farella, "Neural network distillation on iot platforms for sound event detection.," in *Proc. of Interspeech*, 2019, pp. 3609–3613.

[8] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Proc. of ICASSP*. IEEE, 2015, pp. 5206–5210.

[9] Romain Serizel, Nicolas Turpault, Ankit Shah, and Justin Salamon, "Sound event detection in synthetic domestic environments," in *Proc. of ICASSP* , 2020.

[10] Giacomo Ferroni, Nicolas Turpault, Juan Azcarreta, Francesco Tuveri, Romain Serizel, Çagdaş Bilen, and Sacha Krstulović, "Improving Sound Event Detection Metrics: Insights from DCASE 2020," in *Proc. of ICASSP 2021*, 2021.

[11] Francesca Ronchini and Romain Serizel, "A benchmark of state-of-the-art sound event detection systems evaluated on synthetic soundscapes," in *Proc. of ICASSP*, 2022.

[12] Francesca Ronchini, Romain Serizel, Nicolas Turpault, and Samuele Cornell, "The impact of non-target events in synthetic soundscapes for sound event detection," *arXiv preprint arXiv:2109.14061*, 2021.

[13] Francesca Ronchini, Samuele Cornell, Romain Serizel, Nicolas Turpault, Eduardo Fonseca, and Daniel P. W. Ellis, "Description and analysis of novelties introduced in dcase task 4 2022 on the baseline system," in *Proc. of DCASE Workshop*, 2022.

[14] Lu JiaKai, "Mean teacher convolution system for dcase 2018 task 4," Tech. Rep., DCASE2018 Challenge, 2018.

[15] Hongyi Zhang, Moustapha Cisse, and Yann N et al. Dauphin, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[16] Nicolas Turpault, Romain Serizel, Ankit Parag Shah, and Justin Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Detection and Classification of Acoustic Scenes and Events, Workshop, DCASE*, 2019.

[17] Jort F. Gemmeke, Daniel P. W. Ellis, and Dylan et al. Freedman, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. of ICASSP*, 2017.

[18] Justin Salamon, Duncan MacConnell, Mark Cartwright, Peter Li, and Juan Pablo Bello, "Scaper: A library for soundscape synthesis and augmentation," in *Proc. of WASPAA*, 2017.

[19] Frederic Font, Gerard Roma, and Xavier Serra, "Freesound technical demo," in *Proc. ACM*, 2013, pp. 411–412.

[20] Eduardo Fonseca, Xavier Favory, Jordi Pons, and et al. Font, "Fsd50k: an open dataset of human-labeled sound events," *arXiv preprint arXiv:2010.00475*, 2020.

[21] Gert Dekkers, Steven Lauwereins, and Thoen et al., "The sins database for detection of daily activities in a home environment using an acoustic sensor network," in *Proc. of DCASE Workshop*, 2017.

[22] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, "Tut database for acoustic scene classification and sound event detection," in *Proc. of EUSIPCO*, 2016.

[23] Çağdaş Bilen, Giacomo Ferroni, and Francesco et al. Tuveri, "A framework for the robust evaluation of sound event detection," in *Proc. of ICASSP*, 2020.

[24] Nicolas Turpault and Romain Serizel, "Training Sound Event Detection On A Heterogeneous Dataset," in *Proc. of DCASE Workshop*, 2020.

[25] Peter Henderson, Jieru Hu, and Joshua at al. Romoff, "Towards the systematic reporting of the energy and carbon footprints of machine learning," *Journal of Machine Learning Research*, 2020.