Precise Energy Consumption Measurements of Heterogeneous Artificial Intelligence Workloads

René Caspart[1](✉), Sebastian Ziegler[2], Arvid Weyrauch[1], Holger Obermaier[1], Simon Raffeiner[1], Leon Pascal Schuhmacher[1], Jan Scholtyssek[2], Darya Trofimova[2], Marco Nolden[2], Ines Reinartz[1], Fabian Isensee[2], Markus Götz[1], and Charlotte Debus[1]

[1] Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
{rene.caspart,arvid.weyrauch,holger.obermaier,simon.raffeiner, leon.schuhmacher,ines.reinartz,markus.goetz,charlotte.debus}@kit.edu

[2] German Cancer Research Centre (DKFZ), Heidelberg, Germany
{sebastian.ziegler,jan.scholtyssek,darya.trofimova, marco.nolden,fabian.isensee}@dkfz-heidelberg.de

Abstract. With the rise of artificial intelligence (AI) in recent years and the subsequent increase in complexity of the applied models, the growing demand in computational resources is starting to pose a significant challenge. The need for higher compute power is being met with increasingly more potent accelerator hardware as well as the use of large and powerful compute clusters. However, the gain in prediction accuracy from large models trained on distributed and accelerated systems ultimately comes at the price of a substantial increase in energy demand, and researchers have started questioning the environmental friendliness of such AI methods at scale. Consequently, awareness of energy efficiency plays an important role for AI model developers and hardware infrastructure operators likewise. The energy consumption of AI workloads depends both on the model implementation and the composition of the utilized hardware. Therefore, accurate measurements of the power draw of respective AI workflows on different types of compute nodes is key to algorithmic improvements and the design of future compute clusters and hardware. Towards this end, we present measurements of the energy consumption of two typical applications of deep learning models on different types of heterogeneous compute nodes. Our results indicate that 1. contrary to common approaches, deriving energy consumption directly from runtime is not accurate, but the consumption of the compute node needs to be considered regarding its composition; 2. neglecting accelerator hardware on mixed nodes results in overproportional inefficiency regarding energy consumption; 3. energy consumption of model training and inference should be considered separately – while training on GPUs outperforms all other node types regarding both runtime and energy consumption, inference on CPU nodes can be comparably efficient. One advantage of our approach is the fact that the information on energy consumption is available to all users of the supercomputer and not just those with administrator rights, enabling an easy transfer to other workloads alongside a raise in user-awareness of energy consumption.

H. Anzt et al. (Eds.): ISC High Performance 2022 Workshops, LNCS 13387, pp. 108–121, 2022.
https://doi.org/10.1007/978-3-031-23220-6_8

Keywords: Energy measurement · Artificial intelligence · Green AI · Energy efficiency · High performance computing · GPUs

1 Introduction

In the past decade, artificial intelligence (AI) methods have yielded great advances in many areas of science and technology. However, growing complexity in prediction tasks is followed by an equally growing size and complexity in the AI models. Training such large models requires an enormous amount of compute resources, as demonstrated by recent publications [1,9]. In addition, the development process usually includes multiple test runs and hyperparameter optimization, further increasing the needed compute time. While modern accelerator hardware and large-scale computer clusters allow AI researchers to implement such models, the extraordinary need for electricity of these IT-infrastructures poses an increasing challenge, especially with regards to climate change. Recent studies have therefore placed a focus not only on the predictive accuracy of modern AI models, but also on their environmental friendliness in terms of energy consumption and CO_2 footprint [23]. Yet, current efforts rely mainly on estimating electricity consumption from training and prediction (inference) runtimes [26]. Such approaches can only give a rough approximation and do not factor in consumption differences of specific hardware components or executed tasks. To properly gauge the gain in prediction accuracy versus the additional model complexity, as well as raise user awareness on the energy consumption of their AI applications accurate measurements of AI workload energy consumption are needed.

In conventional high-performance-computing, measuring energy consumption of computer code has been investigated thoroughly. Several studies have used either external or internal power meters for assessing the power consumption of commonly used numeric algorithms [5]. For AI models, however, there exists little work on actual measurements of electricity consumption.

Modern deep learning models are increasingly trained on large computer clusters, where measurements via external power meters are not feasible. An alternative is investigating electricity draw of a single device, e.g. a single GPU via NVIDIAs management library (NVML). However, AI workloads are typically run on entire compute nodes, which host nodes with more than one accelerator device or multiple thereof, connected via a fabric. Thus, the energy consumption of the entire training pipeline cannot be precisely captured by linear scaling with the number of GPUs utilized as this would neglect the consumption of the enclosing environment of the accelerator, e.g. CPU, RAM, local disks, fabric, and so forth. Furthermore, despite the tremendous success of GPUs for deep learning applications, access to accelerator hardware is still limited, and many super-computers still host mainly CPU-only nodes.

In order to assess the energy consumption of large-scale neural network training as well as raising user awareness on the carbon footprint of extensive, and potentially inefficient, AI workloads, comprehensive, easily accessible and yet precise assessment of the nodes energy consumption is needed. However, the information on hardware power draw usually requires root access to the system and is therefore not available to common users. Towards this end, the following study presents whole node energy measurements of two use cases representing typical deep learning applications, an image classification problem and a time series forecasting problem. Energy profiles and consumption of these workloads were evaluated in a way that is available to all users of the system. To highlight the differences in heterogeneous hardware compositions, model training and prediction is run on different compute node types with and without GPUs. For all of the experiments, we limit ourselves to measuring the energy consumption in an as-is state of the worker nodes of the HPC cluster. We explicitly do not optimize the node configuration, power limits and CPU frequencies to the specific use case, to imitate the usage scenario of a typical user of an HPC system.

The remainder of the paper is organized as follows: Sect. 2 discusses prior work on the topic of measuring power consumption of compute hardware and energy-efficient AI. Section 3 introduces the use cases, including model architectures and datasets, as well as the compute environment and energy measurement tools utilized in the study. Results of the energy measurements are presented in Sect. 4. Finally, Sect. 5 discusses the found results and future studies.

2 Related Work

Power Aware Computing. Energy Efficient HPC is an important topic for the HPC community, specifically in the light of exascale clusters. Many efforts to study and improve the overall energy efficiency of HPC clusters and corresponding aspects are coordinated and conducted as part of the "Energy Efficient High Performance Computing Working Group" [29].

Many studies are conducted on the energy consumption of HPC systems to guide the design and develop strategies to improve the energy efficiency of an HPC clusters as a whole, e.g. [4,16,18]. Additional studies consider optimizing the energy distribution in an HPC cluster [6,31]. These focus on improving the overall performance of a cluster, while respecting an overall power limit. Patel et al.[20,21] and Shin et al. [24] studied the power consumption and behaviour of an HPC center across many different jobs. Our work shares commonalities with these studies, however, we focus ourselves on AI/ML workflows and aim at providing a view of the energy consumption of typical workloads in this domain, as they are performed by users of HPC clusters on a daily basis. We explore and compare different possible usage options for these jobs on the cluster, aiming to incentivise energy efficiency considerations among the users in this domain.

Several authors have published studies on energy measurements utilizing power meters, which can be categorized into two different approaches: internal or external ones. Among others Suda et al. [27] used external power meters via clamp probes with the aim of verifying a power model for workloads. On their own, these types of measurement are not practical, since their implementation requires substantial efforts and the approach is hardly suited for larger cluster setups, such as high-performance computing clusters. Internal power meters can be further subdivided based on which parts of a system are measured by it. On most nowadays available NVIDIA and AMD GPUs internal power meters are available. These can be read out using high-level libraries and tools, such as NVML [17] or corresponding tools for AMD. Using NVML to provide real-time power measurement data for GPUs has been studied and compared to a proposed power model used for predicting power demands of linear algebraic kernels on GPUs [10]. However, utilizing libraries and tools like NVML yield only power metrics for the GPUs in a system, which makes out only one part of the energy consumption of the full system. Other components such as CPU, memory or local disk, are not taken into account with this approach. Considering the power draw of all components of a node becomes particularly important for scientists having the choice of different compute nodes to run their computation on, e.g. CPU-only nodes and nodes also equipped with GPU accelerators.

Many system vendors are integrating internal solutions for measuring the power demand of a system, which provide important information to HPC operators. A study to make information relying on these tools also available to users of the systems is for example the joint HDEEM project between Bull and Technical University Dresden (TUD), which aims to provide high resolution and accurate power consumption metrics [8]. The approach is also used in production at TUD enabling users to gain information on the energy consumption of their workloads.

Energy Efficiency in AI. Recently, awareness on the energy consumption and eco-friendliness of modern AI methods has been raised [23]. Yet, there are only few reports studying the actual energy consumption of modern day AI algorithms. In general, it is assumed that a reduction in runtime, especially for training, and/or number of parameters results in more energy efficient networks. Several authors rely on estimating power consumption based on the number of used floating point operations per second (FLOPs), e.g. Brown et al. [1]. To reduce training time, authors employ approaches like pre-training and few-shot learning [3]. To reduce the parameter count, sparsity is extensively explored in the literature. So far, these approaches are mostly limited to inference models, i.e. pruning fully trained models to smaller sizes for deployment on low-energy (embedded) hardware, e.g. FPGAs or ASICs [14]. However, direct measurements of the entire energy consumption including all hardware components is rarely performed. Strubell et al. [26] estimated electricity usage and carbon dioxide footprint of training, tuning and inference of several well-known large deep learning models. Their method is based on the runtime of these models, also factoring in the effects of hyperparameter tuning. In an attempt to further raise awareness around the carbon emissions of machine learning methods Lacoste et al. [12] presented a *Machine Learning Emissions Calculator*, that estimates the CO_2 emission of a given model based on the geographical location of the utilized server, the type of utilized accelerator and the overall training time of the model.

Li et al. [13] evaluated the power behavior and energy efficiency of convolutional neural networks (CNN) in commonly used deep learning frameworks on

Table 1. Summary of the computational properties of the use cases *Health* and *Energy*.

Use case	Model	Parameters	FLOPs/sample
Health	CNN	20.4M	10.14G
Energy	LSTM	9.79K	16.13K

both, CPUs and GPUs, namely Intel Xeon CPUs, NVIDIA K20 and Titan X GPUs. Power draw of different CNNs were assessed via Intel's Running Average Power Limit (RAPL) interface for CPU and VRAM [2], and via the NVIDIA System Management Interface for GPUs. Our work is similar to that performed by Hodak et al. [7]. In their study, the authors perform measurements of the total consumed energy as well as relative CPU, GPU and other hardware contributions in a typical image recognition task. They ran training of an ImageNet-based Tensorflow benchmark on multi-GPU-Servers, comprising four 32 GB NVIDIA V100 GPUs and two Intel Xeon Gold 6142 CPUs, and measured both AC and DC draw over the entire AI workload through power meters embedded in the servers power supplies as well as through NVML.

3 Experimental Evaluation

In order to evaluate energy consumption of different AI workloads on heterogeneous hardware nodes, we performed experimental runs of two types of deep learning applications (use cases) on different types of compute nodes on a high-performance computer cluster.

3.1 Workloads

For the use cases, two common types of AI tasks were chosen: a computer vision classification task and a time series regression task. With the aim of measuring energy consumption of AI workloads representing typical scientific applications of deep neural networks, real-world datasets for these two tasks were selected from the research fields *Health* and *Energy*. For both use cases, training and prediction runs with realistic model configurations were conducted on different types of large scale compute nodes, and the overall energy consumption was measured. Table 1 shows a high-level summary of the computational characteristics of the used deep learning models.

Use Case *Energy*. For the use case *Energy*, we chose the task of predicting future electricity consumption (load) over a 7-day period based on historic data. In terms of AI workloads, this corresponds to a classical time-series forecasting, i.e. regression problem. The dataset was derived from the Western Europe Power Consumption Dataset [22], which consists of five years of load data of 15 European countries. The datasets was prepared to be continuous and complete, i.e.

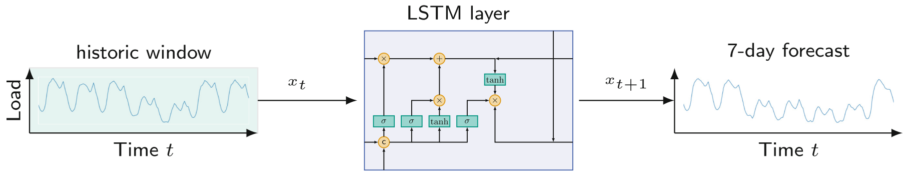

Fig. 1. Schematic LSTM architecture for the use case *Energy* to forecasting electrical load for a 7-day period.

`NaNs` were removed and all load curves were brought to a temporal resolution of 1 h through averaging. Samples were normalized separately for each country to the interval $[0, 1]$. Training data covers the years 2014–2017, validation and test data was taken from the years 2018 and 2019, respectively.

A single layer long-short term memory (LSTM) architecture (cf. Fig. 1) with 48 hidden nodes was used to forecast the hourly electric demand for the next seven days based on the prior seven days load profiles as input [11,15]. The resulting 48 output features were mapped to the required single output feature with one fully-connected layer, i.e. each recurrent loop of the model produces a one-week ahead forecast. While the model itself is rather small in terms of trainable parameters (cf. Table 1) the recurrence in sequence processing results in a substantial computation workload.

The model was trained for 30 epochs with the Adam optimizer at a learning rate of 10^{-3} and a batch size of 64. Loss was calculated as the mean squared error (MSE). All related scripts can be found on GitHub[1].

Use Case *Health*. The second use case *Health* covered the task of predicting a COVID-19 infection based on an lung x-ray images, i.e. an image classification problem. The dataset was taken from the COVID-Net Open Initiative [30] on Kaggle [32]. It comprises 2,358 images of COVID-19-positive patients and 13,993 images of COVID-19-negative patients, collected from various sources. We employed a different data split than the one provided by Kaggle, to prevent data sources in training and test data from overlapping. The training set contains 2,088 positive and 13,696 negative samples, the validation set contains 74 positive and 76 and negative samples and the test set contains 196 positive and 221 negative samples. Images were transformed by applying a logarithmic transform and random blurring. For the prediction model we followed the VGG-19 architecture [25], adding batch normalization and replacing the three fully connected layers in the end by an average pooling and one fully connected layer. The model was trained for 250 epochs using the SGD optimizer with a Cosine Annealing learning rate scheduler at an initial learning rate of 0.1 and a batch size of 64. Data was augmented during training by resizing, applying random horizontal flips and random rotations, taking a random crop of 224 × 224 pixels and finally normalizing the image. For validation and testing the images only

[1] https://github.com/Helmholtz-AI-Energy/AI-HERO-Energy.

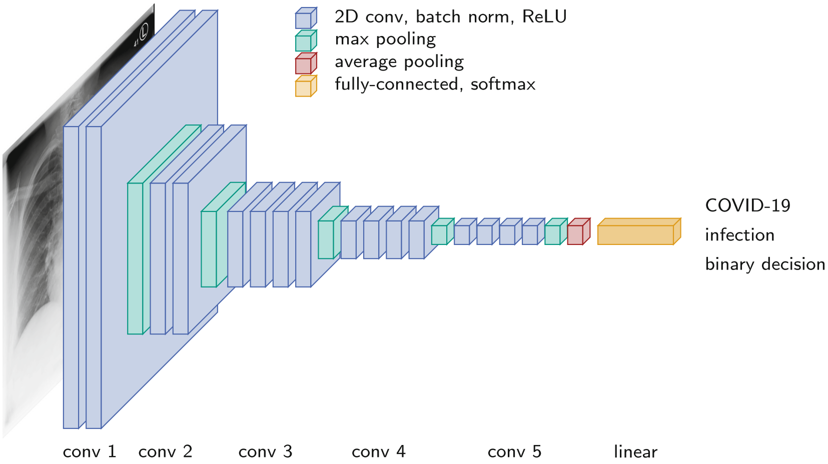

Fig. 2. VGG model architecture for the use case *Health* to predict a COVID-19 infection based on the x-ray input images.

got resized to the respective size and normalized. The entire code used to run the model can be found on GitHub[2].

3.2 Computation Environment

All experiments are conducted on the Tier-2 *HoreKa* supercomputing system, an innovative hybrid cluster with nearly 60 000 Intel Xeon “Ice Lake” processor cores, more than 220 terabytes of main memory, and nearly 700 NVIDIA A100 Tensor Core GPUs. The system is designed as an energy efficient system, peaking at rank 25 in the Green500 [28]. HoreKa consists of two partitions, a CPU-only partition (HoreKa-Blue) designed for highly parallel MPI applications with large memory bandwidth and an accelerated partition (HoreKa-Green) equipped with state-of-the-art accelerators for extremely data- and compute-intensive applications in machine learning. Each of the nodes is a two socket system with Intel Xeon Platinum 8368 CPUs, 38 cores per socket, and two threads per core. It has 64 KB L1 and 1 MB L2 cache per core and 57 MB shared L3 cache per CPU. Horeka-Blue nodes feature 256 GB of main memory and one 960 GB NVMe SSD each. HoreKa-Green nodes are equipped with 512 GB of main memory and four NVIDIA A100-40 GPUs. The operating system of the nodes is Red Hat Enterprise Linux 8.2 with kernel version 4.18.0-193.60.2.el8_2.x86_64, with NVIDIA driver version 470.57.02, and CUDA version 11.4 for the nodes equipped with A100 accelerators. Our use cases are implemented in Python 3.8.0 compiled with GCC 8.3.1 20191121 (Red Hat 8.3.1–5) using the PyTorch framework [19] versioned 1.11.0.dev20210929+cu111. For the interactive access to the compute resources, we utilize the available Jupyterhub service, which uses jupyterlab 3.3.2 and jupyter_server 1.16.0.

[2] https://github.com/Helmholtz-AI-Energy/AI-HERO-Health.

3.3 Measurement Setup

AI workloads, containing the full pipeline of either model training or inference for the two different use cases, were run as batch jobs on the HoreKa system. For measuring energy consumption, we consider four different cases of run setups, depending on the utilized hardware:

- *GPU*: The workload was run as exclusive on one A100 GPU of a HoreKa-Green node, while the other three GPUs were kept idle
- *CPU-mix*: The workload was run on all 76 CPU cores of a HoreKa-Green node, while all of the four GPUs were kept idle
- *CPU-only*: The workload was run on all 76 CPU cores of a HoreKa-Blue node, which do not contain any GPUs
- *Jupyter*: Additionally, an entire analysis pipeline including data exploration and plotting was created in a Jupyter notebook and run on one GPU of a HoreKa-Green node.

Energy consumption of the workloads was assessed via two different sources. For one, internal power sensors of the HoreKa nodes were used to measure whole node energy consumption of the entire workflow. These sensors are part of Lenovo's XClarity Controller (XCC), which can be read via IPMI. To enable access to the energy consumption information without requiring root access on the nodes or sharing of access credentials to the management interfaces of the nodes, a slurm plugin is used. This plugin queries the information from XCC and stores it in slurm's accounting database as accumulated energy consumption. To facilitate a reproducibility of our results and applicability of the method also to other workloads, we rely solely on information which can easily be accessed by any user of the HoreKa system. For the evaluation, we query the average and total energy consumption for the jobs from slurm. As a second source of information, we utilize NVML to assess the individual energy consumption of the GPUs for the workloads *GPU* and *Jupyter* running on accelerator hardware. In order to profile the power draw on the GPUs, NVML was queried every 500 ms. For statistical assessment, runs were repeated five times. We report average measurement parameters for job wall-clock time, average node power draw and overall workload energy consumption.

4 Results

Use Case Energy. The LSTM model achieved a mean absolute percentage error (MAPE) of 5.65% on the unnormalized test dataset within the 30 epochs. Since the test dataset is comparably small and would result in very short inference runtimes with consequently little to no noticeable energy consumption above baseline, measurements of prediction energy consumption were conducted on a separate dataset containing five copies of the training dataset.

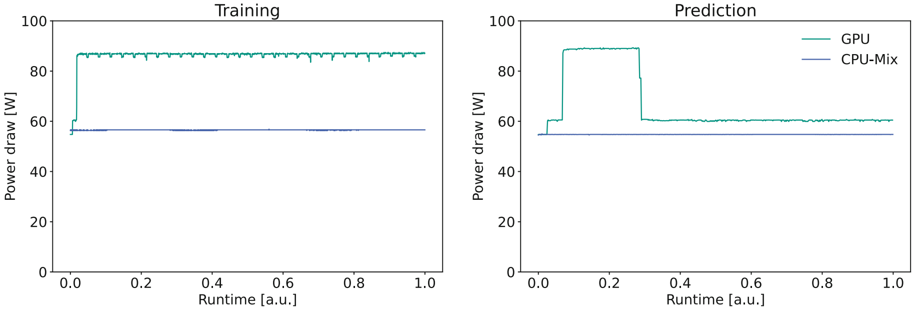

Fig. 3. Jobprofile of the Energy use case, as acquired via NVML.

Table 2. Results of the Energy use case.

	Node	Consumption [kJ]	Average power draw [W]	Runtime
Training	GPU	680.7 ± 6.7	665.8 ± 9.7	00:17:02
	CPU-mix	4 856.0 ± 43.6	644.4 ± 9.9	02:05:37
	CPU-only	2 821.5 ± 70.8	374.1 ± 11.0	02:05:42
Prediction	GPU	156.9 ± 3.4	606.6 ± 8.8	00:04:18
	CPU-mix	320.2 ± 5.9	621.5 ± 7.9	00:08:35
	CPU-only	189.5 ± 4.8	370.6 ± 8.7	00:08:31
Jupyter	GPU	701.9	–	00:17:26

Figure 3 shows the power draw of the LSTM training and inference workload on HoreKa-Green nodes both with (green, *GPU*) and without (blue, *CPU-Mix*) usage of the GPU, as measured by NVML. As expected, when running the model on the nodes CPU-partition, the GPU stagnates at an idle consumption of roughly 55 W. For running the model, the GPU consumes an additional energy of ≈30 W, with small drops between epochs being visible. For prediction, a similar increment in energy consumption can be observed (between 0.05 and 0.3 of the fractional runtime), with the much longer low-energy idle time towards the end of the inference run attributed to result saving.

Results of the overall node energy consumption, average power draw and runtimes of the workload on different node types are given in Table 2. Training of the LSTM network on one NVIDIA A100 GPU is superior to running it on 76 CPU cores with respect to both runtime and energy efficiency: While *GPU* runs consumed only one quarter of the energy the *CPU-only* runs required, they was faster by a factor of ≈7.4. Although the average power draw of the *GPU* runs is almost twice as much as that of the *CPU-only* runs, the immense speed-up achieved through vector processing of the GPU still results in a reduced energy consumption, even for a inherently sequential problem that is a recurrent neural network. Interestingly, while runtimes on were very similar for the *CPU-only* and the *CPU-Mix* runs, the additional idle consumption of the GPUs on mixed nodes

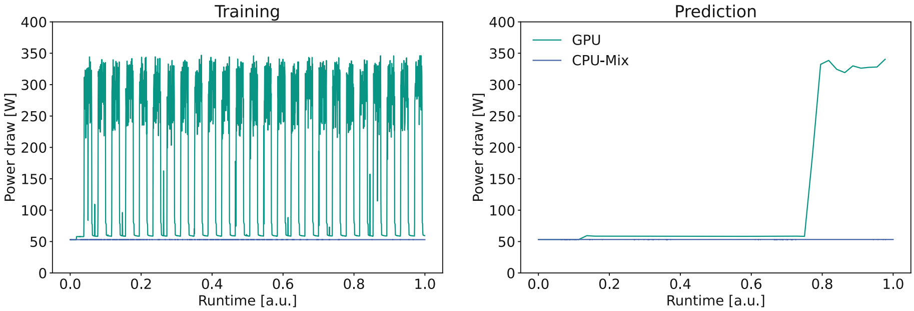

Fig. 4. Jobprofile of the Health use case, as acquired via NVML.

led to a significant increase in energy consumption by a factor of 1.7. Results for the inference runs however show, that even though jobs utilizing the GPUs run faster by a factor of 2, *CPU-only* provides comparable energy consumption. Again, runtimes on both *CPU-only* and *CPU-mix* were comparable, but the additional power draw of the idle GPUs leads to a higher energy consumption of the mixed nodes. Furthermore, we find that running a full analysis pipeline (data exploration, training and inference) in a Jupyter notebook on an A100 of the HoreKa-Green nodes results in similar energy consumption and runtimes as bash processing. However, this is under the assumption, that all cells of the notebook are executed immediately one after another, with no idle-time in between. Since this is usually not the utilization mode of Jupyter notebooks, additional baseline consumption of ≈300 W for notebook idle time will be added for real-world applications.

Use Case Health. The VGG model of the use case *Health* achieved an accuracy of 63.79% on the test set. Training the model to full convergence (250 epochs) took 2 h and 34 min on one A100 GPU, with an overall energy consumption of 7723.958 kJ and an average node power draw of 835.4W. Since running full training on CPUs of an entire node would have taken several weeks to complete, we conducted shortened experiments of 25 epochs to train the VGG model. Due to the small size of the test dataset and the subsequent difficulties in accurately assessing inference power draw, prediction runs were modified such that each sample in the test set was used 10 times for prediction. Results are presented in Fig. 4 and Table 3. The GPU power draw profile exhibits a similar behavior as previously the *Energy* use case: While for the *CPU-Mix* run on mixed nodes the GPU stagnates at an idle consumption around 55 W, the training workload with individual epochs can be clearly seen in the *GPU* run. With this use case however being much more compute intensive due to the processing of images instead of single-value time-series, the additional power draw from the workload amounts to about 300 W on top of the baseline consumption. For prediction, a major fraction of the job runtime was used for data loading, which resulted only in a small increase in energy consumption. The largest contribution to the

Table 3. Results of the Health use case.

	Node	Consumption [kJ]	Average power draw [W]	Runtime
Training	GPU	761.7 ± 17.0	835.6 ± 6.8	00:15:11
	CPU-mix	109 477.1 ± 1 104.9	651.7 ± 7.4	1-22:39:43
	CPU-only	65 796.7 ± 4 899.4	392.1 ± 28.8	1-22:36:31
Prediction	GPU	46.7 ± 2.0	549.5 ± 14.3	00:01:25
	CPU-mix	367.8 ± 10.2	644.5 ± 17.1	00:09:30
	CPU-only	213.5 ± 8.6	377.4 ± 15.6	00:09:25
Jupyter	GPU	783.6	–	00:16:06

power draw budget stems from running model predictions towards the end of the workflow.

Total node energy consumption and runtime of *GPU* runs is superior to runs using only CPUs in training as well as inference, even though the *CPU-only* runs provide a much lower average power draw. Runs on *CPU-only* require about 86 times more energy for training than those on *GPU*, and 4.5 times more energy for inference. The increase in consumed energy of runs on CPUs is not directly proportional to the increase in runtime, since prediction runs on CPUs take ≈ 6.3 times as long as runs on the GPU and training runs took about 194 times as long as on the GPU. Hence the electricity demand of workloads cannot safely be extrapolated from runtime alone, but there is a hardware specific component, making *CPU-only* nodes still relatively efficient in terms of energy consumption. In any case, runs on *CPU-mix* yielded the poorest results with respect to energy consumption as well as runtime.

Running the full training and inference pipeline in a Jupyter notebook results again in similar values for runtime and energy consumption as the batch job on a GPU. The power draw resulting from data exploration and plotting appears to be negligible in comparison to the training workload of the model.

5 Conclusion

In this study, we presented high-precision measurements of whole-node energy consumption of two different AI workloads run on different heterogeneous node types of a large scale supercomputer. Our results show that for image-related deep learning models, running training and inference on a single GPU provides both shorter runtimes and lower power draw than multi-core CPU nodes. The massively parallel processing capabilities of the A100 lead to higher energy efficiency due to the significant reduction in runtime. For non-imaging workloads such as recurrent neural networks for sequential data, inference on CPU yields a comparably low energy consumption as the GPU runs, providing a valid alternative for production runs if there are no runtime constraints.

Our results further demonstrate that energy consumption of composite compute nodes cannot be estimated accurately from linear scaling in runtime of GPU consumption. Especially for sequential data problems, a significant contribution of the energy consumption originates from the baseline of the entire node, e.g. CPU usage and memory access.

From our experiments, it is further evident that GPU idle time results in a non-negligible portion of energy consumption. Hence, GPUs should be utilized for deep learning workflows when available, even if the problem size or network architecture do not demand it straight away. This aspect also makes a strong argument for data parallel multi-GPU training, leveraging the compute power of all accelerators on a node. Finally, we showed that running AI workloads in Jupyter provides comparable energy consumption to submission via batch jobs, thereby facilitating the usage of GPUs and allowing for rapid prototyping while still maintaining energy efficiency.

A major advantage of our approach is the fact that access to metrics of node power consumption measurements is not restricted to users with administration rights, but can be queried by every user of the system for his or her workloads. With this, AI model developers will be sensitized towards the energy footprint of their models and are able to include considerations on energy efficiency into every step of the development process. In future studies, we aim to further map out the energy consumption of different parts of AI workflows through accurately profiling entire node power draw, as well as investigate the energy efficiency of modern AI models, namely self-attention-based architectures. Furthermore, studies taking into account system-level optimization for the power consumption are foreseen.

Acknowledgment. This work was performed on the HoreKa supercomputer funded by the Ministry of Science, Research and the Arts Baden-Württemberg and by the Federal Ministry of Education and Research.

This work is supported by the Helmholtz Association Initiative and Networking Fund under the Helmholtz AI, HIDSS4Health, Helmholtz Imaging and the Helmholtz Metadata Collaboration platform grants and the HAICORE@KIT partition.

References

1. Brown, T., et al.: Language models are few-shot learners. Adv. Neural Inf. Process. Syst. **33**, 1877–1901 (2020)
2. David, H., Gorbatov, E., Hanebutte, U.R., Khanna, R., Le, C.: RAPL: memory power estimation and capping. In: 2010 ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED), pp. 189–194. IEEE (2010)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2018). https://doi.org/10.48550/ARXIV.1810.04805, URL https://arxiv.org/abs/1810.04805
4. Endrei, M., et al.: Energy efficiency modeling of parallel applications. In: SC18: International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 212–224 (2018). https://doi.org/10.1109/SC.2018.00020
5. Ezzatti, P., Quintana-Ortí, E.S., Remón, A., Saak, J.: Power-aware computing (2019)

6. Gholkar, N., Mueller, F., Rountree, B., Marathe, A.: Pshifter: feedback-based dynamic power shifting within HPC jobs for performance. In: Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing, pp. 106–117, HPDC 2018, Association for Computing Machinery, New York, NY, USA (2018). ISBN 9781450357852, https://doi.org/10.1145/3208040.3208047
7. Hodak, M., Gorkovenko, M., Dholakia, A.: Towards power efficiency in deep learning on data center hardware. In: 2019 IEEE International Conference on Big Data (Big Data), pp. 1814–1820. IEEE (2019)
8. Ilsche, T., et al.: Power measurement techniques for energy-efficient computing: reconciling scalability, resolution, and accuracy. SICS Softw.-Intens. Cyber-Phys. Syst. **34**(1), 45–52 (2018). https://doi.org/10.1007/s00450-018-0392-9
9. Jumper, J., et al.: Highly accurate protein structure prediction with AlphaFold. Nature **596**(7873), 583–589 (2021)
10. Kasichayanula, K., Terpstra, D., Luszczek, P., Tomov, S., Moore, S., Peterson, G.D.: Power aware computing on GPUs. In: 2012 Symposium on Application Accelerators in High Performance Computing, pp. 64–73. IEEE (2012)
11. Kong, W., Dong, Z.Y., Jia, Y., Hill, D.J., Xu, Y., Zhang, Y.: Short-term residential load forecasting based on LSTM recurrent neural network. IEEE Trans. Smart Grid **10**(1), 841–851 (2019). ISSN 1949–3061, https://doi.org/10.1109/TSG.2017.2753802
12. Lacoste, A., Luccioni, A., Schmidt, V., Dandres, T.: Quantifying the carbon emissions of machine learning. arXiv preprint arXiv:1910.09700 (2019)
13. Li, D., Chen, X., Becchi, M., Zong, Z.: Evaluating the energy efficiency of deep convolutional neural networks on CPUs and GPUs. In: 2016 BDCloud-SocialCom-SustainCom, pp. 477–484. IEEE (2016)
14. Montgomerie-Corcoran, A., Venieris, S.I., Bouganis, C.S.: Power-aware FPGA mapping of convolutional neural networks. In: 2019 International Conference on Field-Programmable Technology (ICFPT), pp. 327–330. IEEE (2019)
15. Muzaffar, S., Afshari, A.: Short-term load forecasts using LSTM networks. Energy Procedia **158**, 2922–2927 (2019). ISSN 1876–6102, https://doi.org/10.1016/j.egypro.2019.01.952, https://www.sciencedirect.com/science/article/pii/S1876610219310008
16. Nonaka, J., Hanawa, T., Shoji, F.: Analysis of cooling water temperature impact on computing performance and energy consumption. In: 2020 IEEE International Conference on Cluster Computing (CLUSTER), pp. 169–175 (2020). https://doi.org/10.1109/CLUSTER49012.2020.00027
17. NVIDIA Corporation: NVML API Reference. https://docs.nvidia.com/deploy/nvml-api/index.html (2022). Accessed 03 Apr 2022
18. Pakin, S., et al.: Power usage of production supercomputers and production workloads. Concurr. Comput. Pract. Exper. **28**(2), 274–290 (2016). https://doi.org/10.1002/cpe.3191,https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.3191
19. Paszke, A., Gross, S., Massa, F., Lerer, A., et al.: Pytorch: an imperative style, high-performance deep learning library. Adv. Neural Inf. Process. Syst. **32**, 8024–8035, Curran Associates, Inc. (2019). http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
20. Patel, T., Liu, Z., Kettimuthu, R., Rich, P.M., Allcock, W.E., Tiwari, D.: Job characteristics on large-scale systems: long-term analysis, quantification, and implications. In: SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1–17 (2020)

21. Patel, T., Wagenhäuser, A., Eibel, C., Hönig, T., Zeiser, T., Tiwari, D.: What does power consumption behavior of HPC jobs reveal?: demystifying, quantifying, and predicting power consumption characteristics. In: 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 799–809 (2020). https://doi.org/10.1109/IPDPS47924.2020.00087
22. Raucent, F.: Western Europe Power Consumption Dataset. https://www.kaggle.com/datasets/francoisraucent/western-europe-power-consumption (2020)
23. Schwartz, R., Dodge, J., Smith, N.A., Etzioni, O.: Green AI. Commun. ACM **63**(12), 54–63 (2020)
24. Shin, W., Oles, V., Karimi, A.M., Ellis, J.A., Wang, F.: Revealing power, energy and thermal dynamics of a 200pf pre-exascale supercomputer. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2021, Association for Computing Machinery, New York, NY, USA (2021). ISBN 9781450384421, https://doi.org/10.1145/3458817.3476188, https://doi.org/10.1145/3458817.3476188
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
26. Strubell, E., Ganesh, A., McCallum, A.: Energy and policy considerations for deep learning in NLP. arXiv preprint arXiv:1906.02243 (2019)
27. Suda, R., et al.: Accurate measurements and precise modeling of power dissipation of CUDA kernels toward power optimized high performance CPU-GPU computing. In: 2009 International Conference on Parallel and Distributed Computing, Applications and Technologies, pp. 432–438. IEEE (2009)
28. TOP500.org: Green500 List - June 2022. https://www.top500.org/lists/green500/2022/06/ (2022). Accessed 01 July 2022
29. WG, E.H.: Energy Efficient High Performance Computing Working Group. https://eehpcwg.llnl.gov/ (2022). Accessed 01 July 2022
30. Wong, A.: COVID-Net Open Initiative. https://alexswong.github.io/COVID-Net/ (2020). Accessed 29 Mar 2022
31. Zhang, H., Hoffmann, H.: Podd: power-capping dependent distributed applications. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2019, Association for Computing Machinery, New York, NY, USA (2019). ISBN 9781450362290, https://doi.org/10.1145/3295500.3356174
32. Zhao, A.: COVIDx CXR-2 (2021). https://www.kaggle.com/datasets/andyczhao/covidx-cxr2/