Poster Abstract: Energy Efficiency and Timeliness in Model Training for Internet-of-Things Applications

Chih-Shuo Mei
Dept. of Computer Science and Information Engineering
National Taiwan Normal University
Taipei City, Taiwan R.O.C.
60747047s@gapps.ntnu.edu.tw

Chao Wang
Dept. of Computer Science and Information Engineering
National Taiwan Normal University
Taipei City, Taiwan R.O.C.
cw@ntnu.edu.tw

ABSTRACT

Neural network model training is indispensable for domain-specific Artificial Intelligent Internet-of-Things (AIoT) applications. Typically, a GPU graphics card may take several hundreds watts in average during model training, while an embedded GPU device may take only couple watts for the same purpose at the cost of a longer training time. In this paper, we report our empirical study on the model training using NVIDIA RTX 2080 Ti graphics card and NVIDIA Jetson Nano embedded device. We show that, surprisingly, while the training time using the Jetson Nano is 30 times slower than that using the graphics card, the total energy consumption by Jetson Nano is actually only half. The result suggests that when the response time is less critical, one may choose to do model training on GPU embedded devices instead.

CCS CONCEPTS

• **Computer systems organization → Embedded and cyber-physical systems**.

KEYWORDS

Embedded Systems, Energy Efficiency, Deep Learning

ACM Reference Format:
Chih-Shuo Mei and Chao Wang. 2021. Poster Abstract: Energy Efficiency and Timeliness in Model Training for Internet-of-Things Applications. In *International Conference on Internet-of-Things Design and Implementation (IoTDI '21), May 18–21, 2021, Charlottesvle, VA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3450268.3453507

1 INTRODUCTION

GPU-equipped embedded devices have produced a vigorous thrust to Internet-of-Things (IoT) systems research. Industrial Internet reference architecture [2] defines the embedded devices and systems as the *edge tier* and the set of more-capable computing servers as the *platform tier*. Traditionally, the edge-tier devices only collect data for the platform-tier servers to process. Now with the GPU-equipped embedded devices such as the NVIDIA Jetson series, the edge-tier may run AI applications locally. For example, researchers have been studying the use of NVIDIA Jetson TX1 boards for computer-vision workloads for safety-critical system [3]. In this paper, we report our on-going empirical study for the potentials of GPU-equipped embedded devices. In particular, we present the current result of using NVIDIA Jetson Nano embedded device for model training for AI applications.

As we know, from the aspect of computing capacity, there exists a significant gap between the edge tier and the platform tier. But at the same time, edge-tier sub-systems often consume less energy than their platform-tier counterparts. So, for less time-sensitive, domain-specific AI applications, such as predictive maintenance, could it be preferable to perform model training at the edge tier so as to save the overall energy cost? In the following, we present a set of empirical results addressing this question.

Table 1: Hardware Specification

	Jetson Nano	GPU Server
GPU	128-core Maxwell @ 0.92GHz	RTX 2080 Ti @ 1.54GHZ
CPU	Quad-core ARM A57 @ 1.43 GHz	Intel ® Core™i9-9900KF @ 5.00GHz
Memory	4GB LPDDR4	11GB GDDR6

2 EXPERIMENTAL STUDY

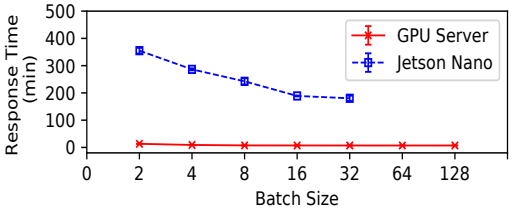
We will call the *GPU server* as our platform tier and *Jetson Nano* as our edge tier. Table 1 lists the specification of the Jetson Nano and the GPU server. Both run Ubuntu Linux 18.04. We train the CNN model based on ResNet18 [1] as our target task, and we compare the power consumption and the response time of the task. For the Jetson Nano, we use the `tegrastats` utility to record the power consumption, and for the GPU server we use the `nvidia-smi` utility.

We study the performance using different batch sizes and the number of epochs. Changing the batch size partitions the training data set differently. With a smaller size of each batch, the number of batches become larger and the workload in each epoch will become heavier. Changing the number of epochs will affect the accuracy for applications that use the trained model. With fewer epochs, the model might be underfitting; with more epochs, the model might be overfitting and at the same time waste both time and energy.

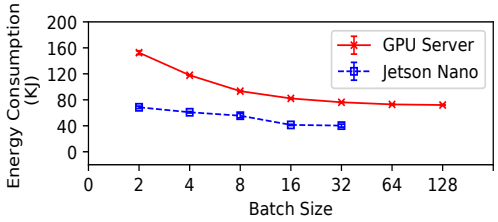
2.1 Observation for Different Batch Sizes

We ran each configuration for five times and for each result we plotted the 95% confidence interval. Figure 1(a) shows that there is significant difference in the response time for model training. For batch size 2, for example, Jetson Nano took about six hours while

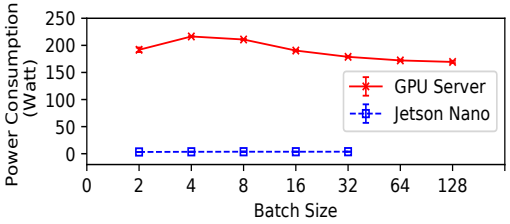
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
IoTDI '21, May 18–21, 2021, Charlottesvle, VA, USA
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8354-7/21/05…$15.00
https://doi.org/10.1145/3450268.3453507



(a) Timeliness of Different Batch Size



(b) Total Energy Consumption with Different Batch Size



(c) Average Power Consumption with Different Batch Size

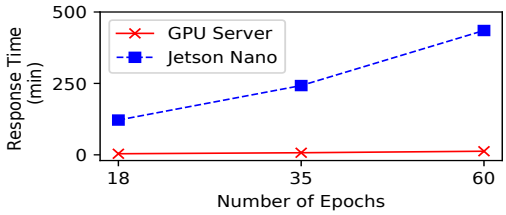
Figure 1: Empirical Result Under Different Batch Size.

the GPU server took eight minutes. But interestingly, the result of total energy consumption is reverse, where the GPU server actually consumed much more energy than Jetson Nano did (Figure 1(b)). while the total energy consumption became stable after batch size 16, the GPU server still cost double energy consumption than Jetson Nano did. Figure 1(a) also shows that the response time became stable after batch size 16, because changing batch size over 16 will make the workload in each epoch become too small to make difference. Jetson Nano can only ran up to batch size 32, and it ran out of memory for larger batch sizes.

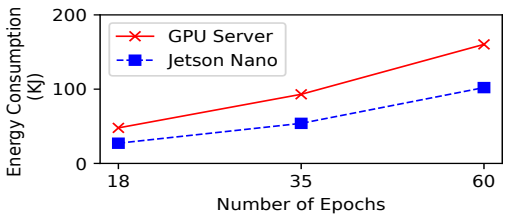
Figure 1(c) shows that the average power consumption of Jetson Nano is much smaller than that of the GPU server under each batch size. The average power consumption is around four Watts for Jetson Nano and around 200 Watts for the GPU server. Notice that while there is significant difference in average power consumption (Figure 1(c)), the total energy consumption for each training is not as large (Figure 1(b)), because the higher response time in Jetson Nano will make its total energy consumption become higher. Nevertheless, in all cases the total energy consumption is still much lower than that of the GPU server.

2.2 Observation for Different Epochs

As shown in Figure 2, increasing the number of epochs will increase the response time, and the change is significant for Jetson Nano.



(a) Timeliness of Different Epochs



(b) Energy Consumption of Different Epochs

Figure 2: Empirical Result Under Different Epochs.

But surprisingly, the total energy consumption of Jetson Nano is still less than that of the GPU server. We ran only up to 60 epochs, because the accuracy converged after 60 epochs.

3 DISCUSSION AND FUTURE WORK

We found that while the response time in the GPU server is several times smaller than Jetson Nano, the energy consumption of the GPU server is almost doubled than that of Jetson Nano. By changing the parameters of model training, we observed that in all cases Jetson Nano outperformed in terms of energy consumption. This indicates that embedded GPU devices like Jetson Nano may be a good choice for less time-sensitive model training. On the other hand, we found that there is a challenge in memory constraint for Jetson Nano when we tried to run the task with batch size over 32. We are going to use a power meter to measure the total device energy consumption of both machines, to complete the picture of this comparative study. Also, we will seek for representative AI applications to demonstrate the applicability of the result.

ACKNOWLEDGMENTS

This work is supported by MOST grant 109-2222-E-003-001-MY3. We would like to thank reviewers to help improve this work.

REFERENCES

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[2] Industrial Internet Consortium. 2019. Industrious Internet Reference Architecture. (Jun 2019). https://www.iiconsortium.org/IIRA.htm
[3] N. Otterness, M. Yang, S. Rust, E. Park, J. H. Anderson, F. D. Smith, A. Berg, and S. Wang. 2017. An Evaluation of the NVIDIA TX1 for Supporting Real-Time Computer-Vision Workloads. In *2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. 353–364.