Forward-Forward: Is it time to bid adieu to BackProp?

Sai Anuroop Kesanapalli
kesanapa@usc.edu

Shashank Rangarajan
sr87317@usc.edu

Anukaran Jain
anukaran@usc.edu

Avtaran Jain
avtaranj@usc.edu

Abstract

The key objective of our project is to implement and study the Forward-Forward (FF) algorithm proposed by Geoffrey Hinton, and compare it with the traditional back-propagation (BackProp) based frameworks. In this report, we discuss the literature surrounding our proposal. We start by examining the works related to FF algorithm and determine their implications on our objective. Next, we probe papers on system performance to construct a viable set of criteria for the evaluation of FF algorithm. Finally, we review the open ended questions, presented by surveying the extant literature, and consider how our proposal addresses them.

We present our literature review[†] in Section 1 and the identified challenges in Section 2.

1 Related Works

In 1.1, 1.2, 1.3, 1.4 we present an overview of recent line of works related to FF in order which help us understand the present status of FF and get an idea about the open challenges that FF poses. In 1.5 we present a few system performance and ML benchmarking works that we draw inspiration from, to aid in our analysis of FF from a practical perspective.

1.1 Forward-Forward

FF is a novel idea and not a lot of work has been done with regards to exploring the potential benefits and drawbacks of using this algorithm. Moreover, there has been a growing interest in the search of alternatives to BackProp, especially in scenarios related to low system analog devices. (Hinton, 2022) explores the feasibility of the FF approach when compared to BackProp and shows that FF performs nearly as well as BackProp when dealing with simple multi layer neural networks. However, the paper does not dive deeper into the pros and cons of this approach, when looking at varying models and architectures. We attempt to reproduce the results demonstrated in the paper, as well as gauge the effectiveness of the proposed approach when dealing with hybrid models.

1.2 Activation Learning

(Zhou, 2022) presents activation learning paradigm as an alternate framework to BackProp, which though is very similar to the FF algorithm proposed by (Hinton, 2022), differs in the way weights are updated. There is an abundance of comparison between activation learning and BackProp present in this work. We intend to refer to these comparisons and formulate our own juxtapositions between FF and BackProp algorithms, and analyze the results for further insights.

1.3 Local Activity Contrastive algorithm

Similarly (Zhu et al., 2022) propose Local Activity Contrastive (LAC) algorithm to learn autoencoders. The idea behind the paper is to use loss functions in every layer of the neural network to replicate locality. Particularly, when learning the difference between activations of two inputs, an original image and an reconstructed image each are minimized. The above is done using two forward passes. LAC is also shown to be beneficial as a pretraining method for Convlutional Neural Networks (CNN), and is what inspired us to delve deeper into a hybrid learning model that uses both BackProp and FF.

1.4 Extensions to FF

Lastly, other studies such as those of (Ororbia and Mali, 2023) utilize FF algorithm as a tool to propose new learning paradigms as generalizations. However, these papers do not perform a direct and detailed analysis between FF and BackProp.

[†] Some of these works are quite new and are available only as a preprint on arXiv.org

1.5 System Performance Metrics and Benchmarking

Benchmarking of Deep Learning (DL) models along with analyzing the system behaviour during the training and inference phases is a crucial step in profiling these models. In many cases, these models run on edge-accelerators which have limited power budget (Khochare et al., 2022). Further, these devices are also constrained by CPU and GPU computational power, hence it is crucial to see how extant and new DL models behave in light of these considerations. Typically, E2E runtime, CPU and GPU utilization and memory consumption (Liu et al., 2019), power and energy consumption (Holly et al., 2020) are logged by several works. (Beutel et al., 2020) report training times, energy, CPU running time with respect to the federated learning framework they propose. (SK et al., 2022) do a detailed system profiling of Nvidia Jetson edge accelerators during the training phase of various deep learning models. In particular, they log E2E time, stall time, GPU compute time, CPU and GPU frequencies, energy consumption, average socket power load as part of their study. However, their work is Jetson-specific and the results may not be universally applicable. While it ambitious to perform a similar fine-grained system analysis for the training phase of FF vs BackProp, such a work will clearly highlight the tangible benefits of FF, or otherwise, rather than focusing on a purely theoretical perspective. MLPerf (Mattson et al., 2020) is a community-driven effort which provides a unified platform for measuring the performance of ML hardware and systems. It is a benchmark suite with one of the major objectives being that it enables a fair comparison between candidate systems. To this end, they assorted a representative set of tasks from several major ML areas, spanning from vision, language, recommendation, to reinforcement learning. Further, for each benchmark, they chose quality metrics close to the state of the art for the corresponding model and data set. While this is a very detailed work, it does not measure low-level system metrics. However, this work sets the tone for creating a similar platform for fair comparison of FF vs BackProp based methods.

2 Challenges

Our detailed review of the literature related to FF algorithm helped identify a few open ended questions.

2.1 What is a nice goodness function to use?

(Zhou, 2022) utilizes a more novel goodness function in their work that focuses on reducing the difference between activations of two forward passes. This suggests that there may be more task-specific goodness functions that can be explored.

2.2 What activation functions can be used?

(Hinton, 2022) uses the ReLU activation in his initial implementation of the FF algorithm, the other viable candidates are unexplored. Choosing the right activation function in FF may heavily influence the performance of a given model. It probably makes sense here to revisit the theory of Boltzmann machines (Hinton et al., 1986) and understand what kind of energy functions are used there.

2.3 Is FF extensible to other architectures like CNNs?

Although (Hinton, 2022) explores this idea to some extent using local receptive fields, it also suspects that a direct implementation of FF algorithm into a CNN architecture may turn out to be problematic. Our initial implementation of FF on a CNN did not yield promising results. This bolsters the aforementioned hunch that the paper had.

2.4 Is FF more efficient than back-propagation, and if so in what scenarios?

Although FF training may be more efficient than BackProp, we do not know if this still holds good during inference phase, as FF encourages very deep architectures that may not entirely fit in memory, especially on the edge-devices where its benefit is supposed to be realized. Our proposed system analysis will shed light into these aspects.

2.5 Can BackProp be augmented with FF to yield better results?

(Zhu et al., 2022) suggest using LAC as a precursor to supervised learning in order to yield better model performance. FF which closely relates to LAC may exhibit similar results when combined with BackProp.

The works related to FF and systems analysis helped us identify the above mentioned challenges and chart an action plan to specifically address the systemic challenges 2.3, 2.4, and 2.5. We presently do not consider the listed theoretical challenges 2.1 and 2.2, leaving it as future work.

References

Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Titouan Parcollet, Pedro PB de Gusmão, and Nicholas D Lane. 2020. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*.

Geoffrey Hinton. 2022. The forward-forward algorithm: Some preliminary investigations. *arXiv preprint arXiv:2212.13345*.

Geoffrey E Hinton, Terrence J Sejnowski, et al. 1986. Learning and relearning in boltzmann machines. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1(282-317):2.

Stephan Holly, Alexander Wendt, and Martin Lechner. 2020. Profiling energy consumption of deep neural networks on nvidia jetson nano. In *2020 11th International Green and Sustainable Computing Workshops (IGSC)*, pages 1–6.

Aakash Khochare, Sai Anuroop Kesanapalli, Rahul Bhope, Yogesh Simmhan, et al. 2022. Don't miss the train: A case for systems research into training on the edge. In *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 985–986. IEEE.

Jie Liu, Jiawen Liu, Wan Du, and Dong Li. 2019. Performance analysis and characterization of training deep learning models on mobile device. In *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 506–515.

Peter Mattson, Christine Cheng, Gregory Diamos, Cody Coleman, Paulius Micikevicius, David Patterson, Hanlin Tang, Gu-Yeon Wei, Peter Bailis, Victor Bittorf, et al. 2020. Mlperf training benchmark. *Proceedings of Machine Learning and Systems*, 2:336–349.

Alexander Ororbia and Ankur Mali. 2023. The predictive forward-forward algorithm. *arXiv preprint arXiv:2301.01452*.

Prashanthi SK, Sai Anuroop Kesanapalli, and Yogesh Simmhan. 2022. Characterizing the performance of accelerated jetson edge devices for training deep learning models. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(3):1–26.

Hongchao Zhou. 2022. Activation learning by local competitions. *arXiv preprint arXiv:2209.13400*.

He Zhu, Yang Chen, Guyue Hu, and Shan Yu. 2022. Contrastive learning via local activity. *Electronics*, 12(1):147.

Contributions

Overall contributions with timeline and status:

Task	Members‡	Timeline	Status
Implement Fully Connected (FC) network using FF	SAK, SR	2/10/23 - 2/24/23	Complete
Study related papers that use FF algorithm	SAK, SR, KJ, AJ	2/25/23 - 3/1/23	Complete
Run FC network using FF on different datasets* and analyze system metrics	SAK, KJ	3/2/23 - 3/24/23	Complete
Implement hybrid (FF + BackProp) approach	SR, KJ	?	Todo
Analyze system performance of hybrid approach	SAK, AJ	4/1/23 - 4/25/23	Todo
Midterm and Final Reports	SAK, SR, KJ, AJ	3/20/23 - 5/3/23	Todo

Contributions specific to this report:

- SAK - Sections 1.5, 2.2 and 2.3
- SR - Sections 1.2, 1.3 and 2.1
- KJ - Abstract, Sections 1.4 and 2.5
- AJ - Sections 1.1, 2.2 and 2.4

Sections and parts of the report not listed above were equally contributed by all the members.

Name Legend:
SAK - Sai Anuroop Kesanapalli,
SR - Shashank Rangarajan,
KJ - Anukaran Jain,
AJ - Avtaran Jain