Forward-Forward Algorithm revisited

2025-2026

Authors
catalin-andrei.manole@s.unibuc.ro
theodor-marian.buciu@s.unibuc.ro
alexandru-ioan.ionescu@s.unibuc.ro

Abstract

This paper is an exploration of the effectiveness and efficiency of The Forward-Forward Algorithm(FFA) and its derivatives compared to clasically trained models. As such we explore the original FFA(Hinton, 2023) as well as PFFA(Ororbia et al., 2023) and EFFA(Kornberg et al., 2023).

1 Introduction

FFA is trying to be more biologically plausible and as a side effect more efficient for certain hardware configurations such as analog hardware(**?**). The algorithm revolves around local update rules based on a goodness score calculated as the $p(positive) = \sigma(\sum y_l^2 - \theta)$ where θ is the hyper-parameter threshold and y_l are the activations of the layer l thus localizing the loss calculation and update on each individual layer thus parallelizing the training process.

More recent papers have begun to tackle some of the problems of FFA. For example the paper "Extending the Forward Forward Algorithm"(Kornberg et al., 2023) found an optimization strategy for the hyperparameter specific to FFA which improves the performance of the network, while other papers like "The Predictive Forward-Forward Algorithm"(Ororbia et al., 2023) have chipped away form the weaknesses of FFA by combining it with predictive coding thus creating a network that is no longer constrained by the the binary classification nature of FFA, as well as introducing a datapath for the propagation of higher level information to lower layers without hurting the local update nature of the original.

In the following, we will try to complete various tasks, comparing FFA to the backprop on both training speed and inferance times. We will try to tackle standard classification problems from simple ones like the XOR-problem to image classification, sentiment analysis, and next token prediction, trying multiple models and comparing performance as well as extending the algorithm our own way.

2 Approach

We are building our neural networks in pytorch, testing the performance of both a Forward-Forward trained model and two traditionally trained models for each task. Having 3 possible hardware configurations(i3-10100F+gtx 1050ti,i7 14650-hx+rtx 4060M, i7-10750H+GTX 1650 ti), we will specify the hardware used for each experiment as to not confuse them. Forward-Forward requires us to have nontraditional architectures since it always trains the model as a binary classifier, forcing us to treat every non-binary classification task as a label matching problem where we give the model all possible combinations of data and label concatenated as input thus being able to classify if the label matches the data. As such one of the traditionally trained models will be identical to the Forward-Forward trained model aside from adding a final layer composed of a single neuron as to replace the goodness score mechanism.

We will test and compare FFA results to the other models by looking at the following metrics:

- Train time
- Number of epochs
- Accuracy
- Memory usage on similar performance
- Overfit and underfit tendencies

2.1 XOR

This is just a validation problem proving that FFA can solve this problem as well as showing difference in the ration of neurons necessary to solve it.

2.2 MNIST

The MNIST dataset is composed of 70,000, labeld 28x28 images of grayscale handwritten digits that is used to benchmark image classification algorithms. For this test we are going to write a FFA MLP network as well as the 2 corresponding MLP backprop networks, and try our hands at a FFA CNN and it's comparison networks .

2.3 IMDB Reviews

The IMDB Reviews dataset consists of 50K tagged movie reviews, the tag can be positive or negative. In this case we are to preprocess the text by removing unwanted data like html tags, stemming them, and using the model word2vec to get the embeddings for the model. Now as for the architecture of the model we are going to use a MLP and it's corresponding backprop models. Here we have the chance to compare the results of this model with a basic RNN or LSTM, these being architectures that are normally impossible to train using FFA.

2.4 Sherlock Holmes Next-Word Prediction Corpus

It is the corpus of Sherlock Holmes. We are going

3 Limitations

This section is mandatory in your report. While we are open to different types of limitations, just mentioning that a set of results have been shown for English only probably does not reflect what we expect. In addition, limitations such as low scalability, the requirement of large GPU resources, or other things that inspire crucial further investigation are welcome.

4 Conclusions and Future Work

This section is mandatory in your report, but it is more informal, don't be afraid to be honest, the evaluation will biased positively by your honesty. Try to cover some of the following aspects:

- now that we did this project, is there anything we could have done different?
- is there any way of improving this project?
- did we like this project or not? (seriously)
- did we learn anything new by doing this?
- suggestions for future projects at this course
- how could things have been more enjoyable?

References

Geoffrey Hinton. 2023. The forward-forward algorithm: Some preliminary investigations. https://www.cs.toronto.edu/~hinton/FFA13.pdf. Lecture notes, University of Toronto.

Jonah Kornberg, David W. Brown, and Benjamin A. Cohen. 2023. Extending the forward forward algorithm. *arXiv preprint arXiv:2307.04205*.

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