Evaluation of the Forward-Forward Algorithm for Sentiment Classification

Joseph Edell
New York University
jre6163@nyu.edu

1. Introduction

Sentiment classification, like many tasks within the field of machine learning, relies heavily on the use of backpropagation algorithm [10] [8] to learn features and predict the sentiment of a given text. This algorithm efficiently computes the gradients of the weight parameters in a neural network by propagating the error backwards from the loss function to each layer. However, backpropagation falls short as a model of the way in which the brain learns, and for use on efficient analog hardware. Thus, the objective of this study is to explore an alternative algorithm to backpropagation, namely, the Forward-Forward algorithm [3], in the context of the sentiment classification task.

The present evaluation expands on the original proposal by implementing the Forward-Forward algorithm using transformer encoders. Additionally, an effective method for generating positive and negative data from text is also explored. Finally, a comparison is made between the training performance of the Forward-Forward algorithm and traditional backpropagation under identical model parameters and architectures.

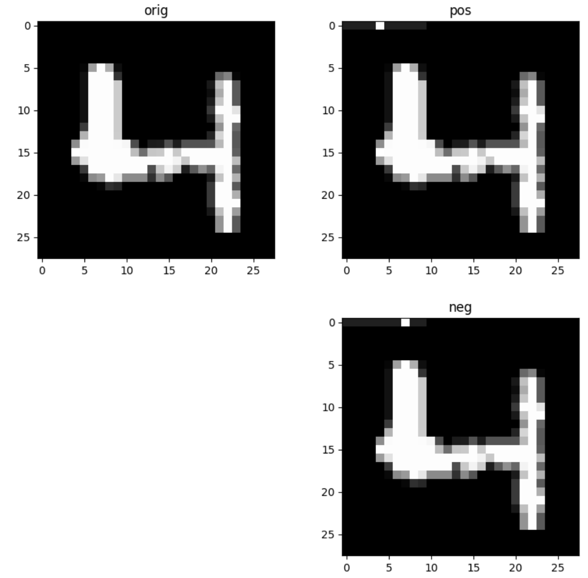

Figure 1. Examples of positive and negative data with labels overlayed in the first 10 pixels from MNIST [6]

2. Background

2.1. Problems with Backpropagation

The Forward-Forward algorithm, introduced by Hinton in December 2022, is a learning procedure for neural networks that replaces the forward and backward passes of backpropagation with two forward passes, one with positive data and the other with negative data. Hinton argues that the perceptual system needs to perform ”inference and learning in real time” without the need to take a timeout to update the internal state of the model. In his proposal, the Forward-Forward algorithm offers a more biologically plausible model of how the brain learns as opposed to backpropagation, which requires a computationally intensive backwards pass and complete knowledge of the computations performed during the forward pass to propagate the gradient information to each layer in the model.

Currently, real-time learning of deep neural networks is mainly conducted via reinforcement learning algorithms, which can suffer from high variance and scalability issues when applied to networks with a large number of parameters. The Forward-Forward algorithm has the potential to address these challenges and produce a more accurate model of how brains learn by passing data through the network without storing information between layers or stopping to propagate gradients.

2.2. Forward-Forward

The Forward-Forward replaces the forward and backward passes of backpropagation through the utilization of two forward passes that operate identically, but on different data and with opposite objectives. Specifically, the algorithm performs a positive pass on real data to enhance the goodness of each hidden layer by adjusting the weights, followed by a negative pass on fake data to reduce the goodness of each hidden layer through the same weight adjustments.

The aim of the algorithm bears resemblance to the concepts employed in contrastive learning methodologies [2]. In contrastive learning, a similarity metric $E(X_1, X_2)$ is assigned to pairs of inputs, and the objective of the system is to minimize the energy for positive input pairs (i.e., inputs belonging to the same class) while simultaneously maximizing the energy for negative input pairs (different classes). A contrastive loss function can be defined as follows:

$$\begin{aligned} L(Y, X_1, X_2) = (1 - Y)L_P(E(X_1, X_2)) \\ + Y L_N(E(X_1, X_2)) \end{aligned}$$

where Y is the label (positive pair or negative pair) and L_P and L_N are partial loss functions for positive and negative pairs respectively.

The aim of each layer within the Forward-Forward algorithm is to maximize the level of ”goodness” for positive data above a predetermined threshold and to minimize the ”goodness” for negative data below this threshold. One possible measure of goodness is the summation of the squared activities in a layer, though other measures may be employed. In this context, we denote the measure of the sum of squared activities in a layer for input X as follows:

$$G(X) = \sigma\left(\sum_j F(X)_j - \theta\right)$$

The notation $F(X)$ denotes the output of the layer with index j, while σ refers to the logistic function and θ denotes a predetermined threshold value. The loss function associated with this objective can be expressed in the following manner:

$$L_{pos}(X_{pos}) = \log(1 + exp(-G(X_{pos})))$$

$$L_{neg}(X_{neg}) = \log(1 + exp(G(X_{neg})))$$

$$L(X_{pos}, X_{neg}) = L_{pos}(X_{pos}) + L_{neg}(X_{neg})$$

Specialized partial loss functions with opposing objectives have been designed for both contrastive learning and the Forward-Forward algorithm, tailored to the specific nature of the input they receive. The ultimate goal of both methods is to minimize the loss associated with their respective positive samples.

Compared to backpropagation, Forward-Forward offers several potential advantages. One such advantage is that it enables the insertion of a ”black box” model between the layers trained with Forward-Forward, which may lead to improved system performance against data perturbations in the long run. Additionally, the positive and negative passes may potentially be performed separately without sacrificing performance. Learning is simplified in the positive pass, enabling a continuous stream of information to be streamed through the network without the need to store activities or pause for derivative propagation. In contrast, a negative pass can be carried out during an offline ”sleep” phase. Backpropagation, however, is often incompatible with high levels of model parallelism and limits potential hardware designs [7].

2.3. Transformers

The Transformer was introduced by Vaswani et al. [9]. This architecture is primarily used for natural language processing tasks, such as machine translation and text summarization, and many variants exist for a variety of natural language understanding tasks. It consists of a stack of encoder and decoder layers, each of which contains multi-head self-attention and position-wise feedforward sublayers. The self-attention mechanism allows the network to weigh different parts of the input differently based on their relevance to the task at hand, while the feedforward sublayer applies a pointwise nonlinear transformation to the input. The overall architecture of the Transformer has been shown to be highly effective at capturing long-range dependencies in sequential data.

This study offers an implementation of the Forward forward algorithm that performs layer wise gradient updates on transformer-encoder blocks.

3. Overview

The aim of this study is to investigate alternative approaches to backpropagation in natural language understanding (NLU) tasks. Specifically, the study seeks to evaluate the Forward-Forward algorithm proposed by Hinton [3] as a substitute for backpropagation. Although Hinton’s initial evaluations demonstrate promising results in a range of tasks and datasets, such as image classification with MNIST [6] and CIFAR-10 [4] as well as sequence prediction, this study seeks to assess the algorithm’s efficacy in sentiment classification tasks and compare it to conventional models. Additionally, the project investigates methods of generating negative data for NLU tasks and examines possible architectures for this alternative to backpropagation.

4. Implementation

4.1. Details of the Architecture

This investigation leverages the Forward-Forward algorithm by utilizing transformer-encoder and linear blocks with local gradient updates. Each block possesses its own local loss function, and the gradient update is performed

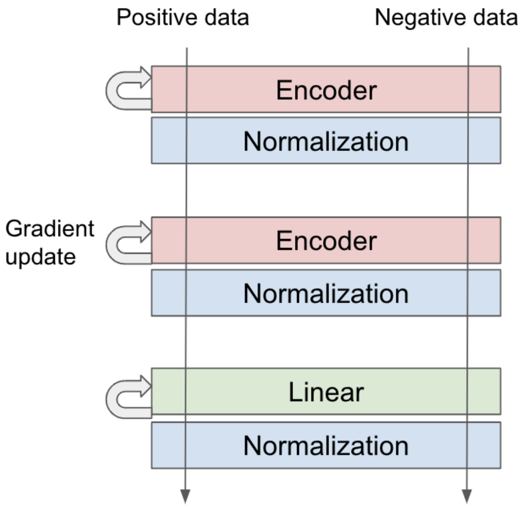

Figure 2. Visualization of the layer-wise local gradient update implemented in this study.

layer-wise. Before being passed to the encoder blocks, the data undergoes positional encoding, and each encoder block consists of multi-headed self-attention and point-wise feed-forward sublayers. The final layer of the network is a simple fully-connected layer.

4.2. Normalization

In accordance with the Forward-Forward algorithm, normalization is applied between each block in the network. The purpose of this normalization is to prevent the sharing of information about the length of the activity vector between layers. If this information were shared, distinguishing positive from negative data would be simple, as the network could use the lengths of the vectors rather than learn information from their relative orientation. The normalization formula implemented is as follows:

$$\text{Norm}(x) = \frac{x}{||x||_2}$$

where the second term is the L_2 norm of x. This formula is a simplified version of layer normalization [1].

5. Results

This study aimed to evaluate the Forward-Forward algorithm as an alternative to backpropagation in the context of natural language understanding tasks. The algorithm used transformer-encoder and linear blocks with local gradient updates, and evaluated its performance on sentiment classification tasks. The results of our experiments, shown in Figure 3, indicate that while the Forward-Forward algorithm can achieve reasonable accuracy on the tasks, it is significantly slower than backpropagation. This observation is consistent with Hinton's proposal and other works that have used similar local gradient updates, such as Ren et al. [7].

Model	**Transf-Encoder**
Epochs	**200**
Metric	Test / Train Err. (%)
Backprop	30.56 / 18.98
Forward-Forward	**45.32 / 42.57**

Figure 3. Results of sentiment classification task

Further exploration of potential modifications to the algorithm may lead to improved performance and efficacy in NLU tasks. Additionally, the work on generating contrastive-like samples for text classification and investigating viable architectures for the Forward-Forward algorithm may contribute to future research efforts in this area.

6. Next Steps

This study is a preliminary step into research into more biologically plausible models of how networks can learn and perceive the world. Further research should explore the difference in learning speeds between local gradient based methods and backpropagation, experiment with differing activation, loss, and goodness functions, as well as apply the Forward-Forward algorithm to other NLP tasks, particularly next-token prediction.

It is possible to draw parallels between the Forward-Forward algorithm and recently proposed models of the world that can plan hierarchically and take actions based on an internal representation of sensory data [5]. Such models passively observe their environment to learn how it functions and to take actions based on an internally constructed model. By applying the Forward-Forward algorithm to learning these representations, it is possible to achieve continuous online learning from positive observed data, with a ”sleep” phase used to learn from negative data (potentially generated offline from positive data). This approach is comparable to how humans learn to navigate and act within their environment.

7. Conclusion

This study evaluated the effectiveness of the Forward-Forward algorithm, an alternative to backpropagation, on sentiment classification tasks using transformer encoders. The implementation proposed was evaluated against an identical model using backpropagation, and the results showed that while the Forward-Forward algorithm can learn with transformer-encoder blocks with local gradient updates, learning is far slower than with backpropagation. This finding corresponds with Hinton's proposal and other works using similar local gradient updates. Further research should explore the difference in learning speeds between local gradient-based methods and backpropagation, experiment with differing activation, loss, and goodness functions, and apply the Forward-Forward algorithm to other NLP tasks, particularly next-token prediction. Overall, the Forward-Forward algorithm represents a promising direction for research into more biologically plausible models of how networks can learn and perceive the world, with potential applications in fields such as robotics and autonomous systems.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. 3

[2] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1, 2005. 2

[3] Geoffrey Hinton. The forward-forward algorithm: Some preliminary investigations, 2022. 1, 2

[4] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009. 2

[5] Yann LeCun. A path towards autonomous machine intelligence. Version 0.9.2, 2022-06-27, 2022. 3

[6] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010. 1, 2

[7] Mengye Ren, Simon Kornblith, Renjie Liao, and Geoffrey Hinton. Scaling forward gradient with local losses, 2023. 2, 3

[8] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. 1986. 1

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. 2

[10] Paul Werbos and Paul John. Beyond regression : new tools for prediction and analysis in the behavioral sciences /. 01 1974. 1

Model	Blocks	d_{model}	d_{hidden}	Dropout	Epochs	Learning Rate	Momentum
Encoder-Only FF	6	128	512	0.1	200	1e-3	0.9
Encoder-Only Backprop	6	128	512	0.1	200	1e-3	0.9

Table 1. Architecture details