Imperial College London
Department of Earth Science and Engineering
MSc in Applied Computational Science and Engineering

Independent Research Project
Project Plan

A Spiking Neural Network-Based Implementation of Forward-Forward Algorithm

by

Boxuan Zhu

Email: boxuan.zhu22@imperial.ac.uk
GitHub username: acse-bz422
Repository: https://github.com/ese-msc-2022/irp-bz422

Supervisors:
Dr Dan Goodman
Dr Lluis Guasch

June 2023

Abstract

Spiking neural networks inspired by biological neurons have difficulties in network training. This project aims to implement an algorithm called “forward-forward” in spiking neural networks to alleviate their training difficulties. The forward-forward algorithm is a learning algorithm that does not require backpropagation and allows each hidden layer to locally optimize the weights.

Introduction: Literature Review

Compared with traditional artificial neural networks, the neurons of spiking neural networks are more similar to biological neurons. Spiking neural networks have two distinct features that bring both advantages and disadvantages to SNNs:

a. Spiking neural networks include a temporal element. During calculation, each neuron of the spiking neural network will accumulate input spikes (discrete values) in the time dimension, and will be excited and output spikes when a certain threshold is reached. This makes the spiking neural network inherently advantageous for sequential data processing. On the other hand, the calculation mode of SNN is also distinguished from the mode of processing continuous values of traditional ANN, and makes the calculation overhead of SNN significantly lower than that of traditional ANN (Maass, 1997).

b. Spiking neural networks are not differentiable. The excitation mechanism of a spiking neural network makes it impossible to perform gradient calculations during training - the excitation function of a spiking neural network is inherently non-differentiable. This brings great challenges to the training and practical application of spiking neural networks (Neftci, Mostafa, & Zenke, 2019).

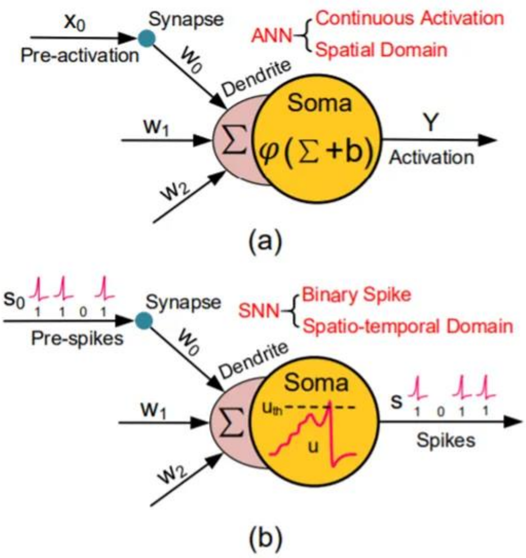

Fig 1. Comparison of traditional neurons and spiking neurons (Deng et al., 2020)

Currently, there are two main implementations of spiking neural networks: "ANN to SNN" and "ANN+SNN". "ANN to SNN" means to directly transplant the trained neural network into SNN to avoid the training difficulty of SNN. "ANN+SNN" means adding a part of the SNN structure to the ANN to obtain the advantage of low power consumption of the SNN (Zhu, Zhao, & Eshraghian, 2023).

Compared with the traditional training algorithm, the forward-forward algorithm abandons backpropagation, an indispensable method in modern machine learning, and uses two forward propagations for weight correction (Hinton, 2022). Specifically, the two forward passes have the same calculation mode, but use opposite data (positive data and reverse data), and their objectives (goodness) are also opposite. Eventually, the model trained by the forward-forward algorithm will reach a state where each layer of neurons in the network has high enough goodness for positive samples and low enough goodness for negative samples.

Similarly, the forward-forward algorithm also has two distinct characteristics:

> a. Forward-forward algorithms do not need to perform backpropagation. A neural network trained using the forward-forward algorithm essentially allows each hidden layer to have its own "loss function" and complete their respective training independently. This makes the hidden layers no longer depend on each other. Even developers can put some kind of black box (noise or perturbation) between layers to improve robustness (Hinton, 2022).
>
> b. Forward-forward algorithms have advantages for sequential data processing. Since there is no need to repeatedly stall during training for backpropagation, the forward-forward algorithm can infer and learn in real time (Hinton, 2022).

It can be found that both the spiking neural network (Mozafari, Kheradpisheh, Masquelier, Nowzari-Dalini, & Ganjtabesh, 2018) and the forward-forward algorithm have advantages in sequential data processing and have the potential for real-time reasoning and learning. More importantly, the problems caused by the non-differentiable characteristic of the spiking neural network can be partially solved by the forward-forward algorithm (see the next part for details). Considering the common and complementary characteristics of the two technologies, the spiking neural network and the forward-forward algorithm, the combination of the two is feasible.

Introduction: Problem Description

The key to this project is to solve the following problem: how to successfully train on non-differentiable spiking neural networks by exploiting the property that the forward-forward algorithm does not require backpropagation. This problem can be broken down into two sub-problems. First, although the forward-forward algorithm no longer uses backpropagation, it still performs gradient calculations within each hidden layer - this gradient calculation is based only on the current layer's own "goodness" function. This is still a serious hindrance to spiking neural networks. Second, assuming we implement a new algorithm that completely abandons backpropagation and gradient calculations, this algorithm still requires further modifications - the excitation mechanism and information encoding method of the spiking neural network are different from those of the traditional artificial neural network (Guo, Fouda, Eltawil, & Salama, 2021).

Introduction: Key idea

Spiking neural networks are difficult to train. But algorithms using the new mechanism have the potential to successfully train spiking neural networks. These algorithms need to meet two key conditions: 1. Backpropagation is not performed to ensure that the spiking neural network can reason and learn in real time. 2. Gradient calculation is not performed to avoid the non-differentiable characteristic of spiking neural networks.

In this project, the forward-forward algorithm can free the spiking neural network from backpropagation and avoid part of the gradient calculation (chain rule in backpropagation). Also, by using some special algorithms, the remaining gradient calculation (each hidden layer still has its own gradient calculation) can also be circumvented.

Introduction: Objectives

Based on the above problems, the objective of this project (need to be completed sequentially) is:

1. Improve the forward-forward algorithm so that backpropagation and gradient calculation are no longer required.

2. Implement the improved forward-forward algorithm in the spiking neural network. This includes the following subobjectives:

    2.1. Adapt the algorithm to handle spikes in the input data.

    2.2. Transform the algorithm and add the neurodynamic equations and special structures required by the spiking neural network.

3. Using a range of data sets, test and evaluate the combination, including performance evaluation and robustness evaluation.

If objective 1 cannot be fully achieved - the improved algorithm still requires gradient calculations, a viable alternative is to replace the activation function of the spiking neural network with a differentiable approximation function when achieving objective 2.

Progress to Date

- Forward-forward algorithm related paper reading and code case studies

- Readings and code case studies on spiking neural network related papers

- Analysis of genetic algorithm (whether it can be used in the feasibility analysis of this project)

- Analysis of reinforcement learning (whether it can be used in the feasibility analysis of this project)

- Exploration of machine learning methods without gradient calculation and related paper reading

Progress on subproblem 1: How to make new algorithms in spiking neural networks not use backpropagation and gradient calculation?

> First, the forward-forward algorithm can already be trained without backpropagation, so the problem to be solved is the gradient calculation in each hidden layer. Three approaches to this problem have been identified: 1. Use random perturbations (genetic algorithm and particle swarm optimization) to optimize the weights, thereby avoiding the use of gradient computations. 2. Use alternative gradients (approximate functions from non-differentiable activation functions) 3. Use the unique R-STDP algorithm in SNN for weight optimization.

Progress on Subproblem 2: How to make the forward-forward algorithm able to handle spike-shaped data?

> The solution to this problem involves the selection of a series of frameworks and models: 1. Choose a spike-based information encoding framework. 2. Choose a spiking neuron model (such as Integrate-and-Fire model and Leaky Integrate-and-Fire model). The analysis of this sub-question was still ongoing at the time of completion of this report.

Future Plan

1. General plan

Phase 1: Reproduce the forward-forward algorithm in the traditional artificial neural network and try to improve this algorithm so that it does not need to use gradient calculations.

Phase 2: Further transform the forward-forward algorithm and port it into the spiking neural network.

Phase 3: Evaluate the spiking neural network. Various datasets will be used to evaluate its performance and robustness.

2. Mitigation plan

Mid-June - July 1 (or earlier): Phase 1.

July 1 (or earlier) - August 1: Phase 2. If this phase is not completed before August 1, it will be postponed to August 10.

August 1 - August 10: Backup time for the most challenging porting process.

August 10 (or earlier) - August 20: Phase 3. Complete the final report. Complete presentations.

August 20 - August 31: Backup time for completing final report and presentations.

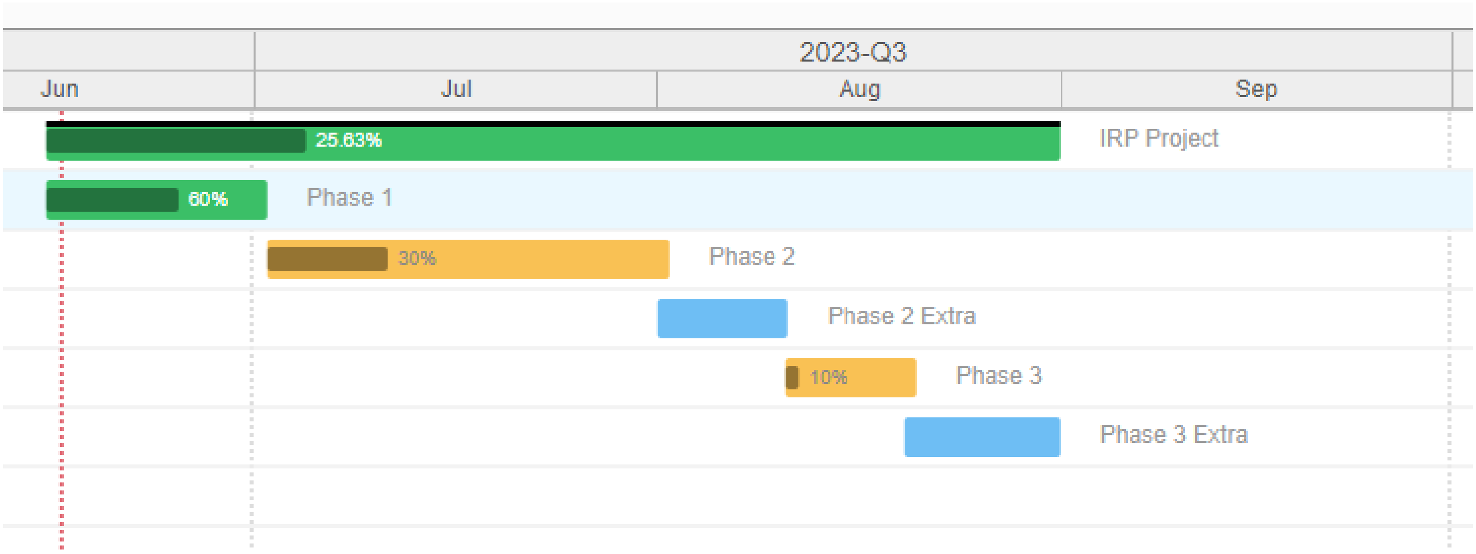

Fig 2. IRP Gantt chart

Precautions: Failed to avoid gradient computation

If the final algorithm is not available by August 1st, use an alternative gradient (approximate function) in the SNN to continue the project development.

References

Deng, L., Wu, Y., Hu, X., Liang, L., Ding, Y., Li, G., . . . Xie, Y. (2020). Rethinking the performance comparison between SNNS and ANNS. *Neural networks, 121*, 294–307.

Guo, W., Fouda, M. E., Eltawil, A. M., & Salama, K. N. (2021). Neural coding in spiking neural networks: A comparative study for robust neuromorphic systems. *Frontiers in Neuroscience, 15*, 638474.

Hinton, G. (2022). The forward-forward algorithm: Some preliminary investigations. *arXiv preprint arXiv:2212.13345*.

Maass, W. (1997). Networks of spiking neurons: the third generation of neural network models. *Neural networks, 10*, 1659–1671.

Mozafari, M., Kheradpisheh, S. R., Masquelier, T., Nowzari-Dalini, A., & Ganjtabesh, M. (2018). First-spike-based visual categorization using reward-modulated STDP. *IEEE transactions on neural networks and learning systems, 29*, 6178–6190.

Neftci, E. O., Mostafa, H., & Zenke, F. (2019). Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine, 36*, 51–63.

Zhu, R.-J., Zhao, Q., & Eshraghian, J. K. (2023). Spikegpt: Generative pre-trained language model with spiking neural networks. *arXiv preprint arXiv:2302.13939*.