

Towards DNN Training at the Edge with Direct Feedback Alignment

Matteo Lai¹, Davide Nadalini² and Francesco Ratto¹
¹Università degli Studi di Cagliari, ²Università di Bologna
 francesco.ratto@unica.it

Abstract Training neural networks at the edge enables self-adaptive and evolving systems through on-device and federated learning. However, memory and computational constraints make such training approaches, often based on backpropagation, challenging, particularly on MicroController Units (MCUs). In this work, we investigate the deployment of on-device training, based on **Direct-Feedback Alignment (DFA)**, a biologically plausible method that replaces weight-symmetric error propagation with fixed random feedback connections. We present a the first **implementation of DFA** on edge devices at the state-of-the-art, targeting Parallel Ultra-Low Power (**PULP**) MCUs. Our implementation is based on the PULP-TrainLib framework. Experiments on the MNIST dataset demonstrate the feasibility of DFA-based training on resource-constrained devices, achieving competitive accuracy, latency and memory usage compared to standard approaches. We further discuss strategies for **latency and memory optimization**, including sparse update and buffer reuse, and outline a path toward resource-efficient deployment of DFA-based training on edge ultra-low-power MCUs.

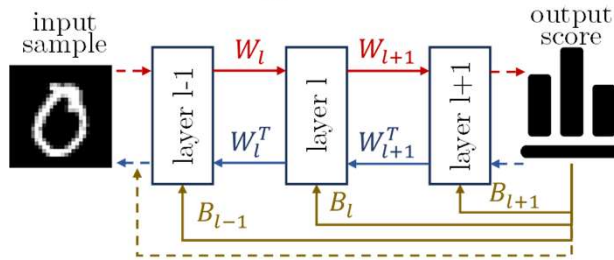


Motivation

- **Accuracy degradation under domain shifts:** deployed DNNs on edge devices remain static after training.
- **Resource constraints of on-device learning:** backpropagation (BP) learning is often too memory- and compute-intensive for resource-limited MCUs.
- **Need for lightweight training tailored to low-energy hardware:** there is a clear demand for efficient on-device learning approaches designed for ultra-low-power MCUs.

Direct Feedback Alignment

With **DFA training**, given a DNN composed of fully-connected layers, the error δ_l at layer l is directly derived from the **error of the entire network** δ_{output} through a fixed, random **noise matrix** B_l of appropriate size: $\delta^l = \frac{\partial L}{\partial a^l} \circ \sigma'(z^l) = B^l \cdot \delta_{\text{output}} \circ \sigma'(z^l)$

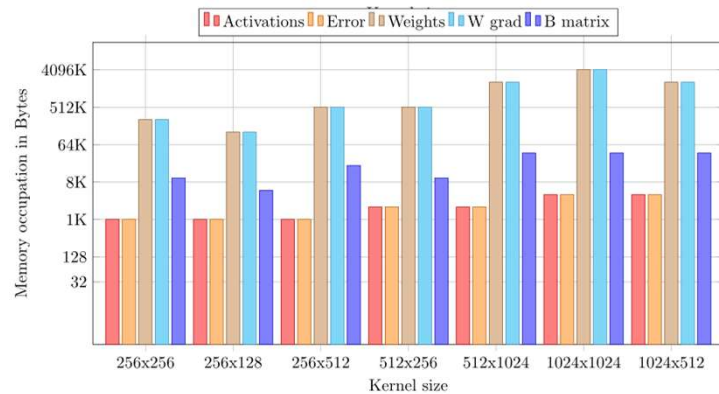
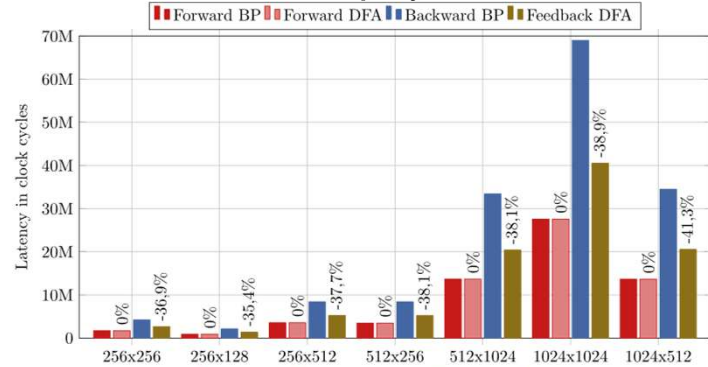


- **Forward pass:** executed for both training methods
- **Backward pass:** executed for BP training
- **Feedback pass:** executed for DFA training

Implementation and Preliminary Result

- **Benchmark** our DFA-based on-device learning implementation on a PULP SoC, a 10-core RISC-V-based ultra-low-power platform with hierarchical memory and DSP extensions. We leverage **PULP-TrainLib** for baseline **hardware-optimized training kernels for BP** and **GVSoc simulator** to profile the execution.
- **Implement DFA training kernels** for fully connected layers on PULP-TrainLib's FP32 primitives, introducing **optimized feedback step functions** that use parallelized matrix multiplication, store random feedback matrices in L2 memory, and reuse the available SGD optimizer.

Comparison of BP and DFA over single layers in terms of latency and memory occupation.



Comparison of main computation and memory for baseline BB and DFA training.

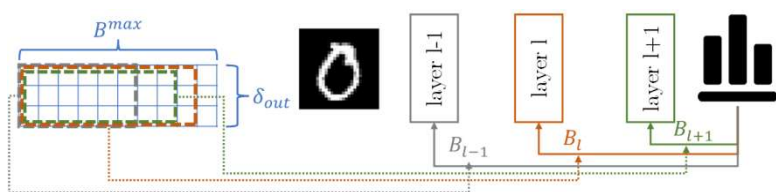
Fully conn. layer: $a^l = \sigma(W^l \cdot a^{l-1})$	Forward	Backward/Feedback	Memory
Backpropagation	$W^l \cdot a^{l-1}$	Error: $(W^{l+1})^T \cdot \delta^{l+1}$ W grad: $\delta^{(l)} \cdot (a^{(l-1)})^T$	$z^l \in R^B, \frac{\partial L}{\partial a^l} \in R^B, W \in R^{A \times B}, \frac{\partial L}{\partial W} \in R^{A \times B}$
Direct Feedback Alignment	$W^l \cdot a^{l-1}$	Error: $B^l \cdot \delta^{out}$ W grad: $\delta^{(l)} \cdot (a^{(l-1)})^T$	$z^l \in R^B, \frac{\partial L}{\partial a^l} \in R^B, W \in R^{A \times B}, \frac{\partial L}{\partial W} \in R^{A \times B}, B \in R^{A \times O}$

Comparison of BP and DFA over a 2-layer DNN for MNIST classification trained with Biotorch [1].

Training Method	Accuracy 15 epochs	Forward Latency 1 Core	Forward Latency 8 Cores	Back./Feedb. Latency 1 Core	Back./Feedb. Latency 8 Cores	Memory Occup.
BP	98.2	24.8K cc	3.9K cc	34.5K cc	7.1K cc	242 MB
DFA	97.9	24.6K cc	4.0K cc	34.5K cc	6.9K cc	246 MB
Variation	+0.3%	-0.5%	+1.0%	-0.1%	-2.7%	+2.1%

Envisioned DFA Optimizations

DFA with a unique shared noise matrix.



- **Unique Noise Matrix:** a single global feedback matrix can be shared across all DFA layers by slicing it per layer [2].
- **Memory Buffer Sharing for Activations:** a single memory buffer can be shared for all layers by recomputing activations on-the-fly during weight updates
- **Sparse Weight Update and Layer Freezing:** reducing computation in on-device learning, with sparse updates modifying only sensitive weights and layer freezing selectively updating layers [3], [4].

[1] Sanfiz, A. et al. 'Benchmarking the accuracy and robustness of feedback alignment algorithms.' arXiv preprint arXiv:2108.13446 (2021).
 [2] Julien Launay et al. "Principled training of neural networks with direct feedback alignment". arXiv:1906.04554 (2019).

[3] Shuai Zhu et al. "On-device training: A first overview on existing systems". In: ACM transactions on sensor networks 20.6 (2024).
 [4] Brian Crafton et al. "Direct feedback alignment with sparse connections for local learning". In: Frontiers in neuroscience 13 (2019).