Adapting Mono-Forward with Zeroth-Order Gradient Estimation for Automatic Differentiation-Free Training

Ateş Görpelioğlu **Supervisors**: Stephanie Tan, Yaqi Guo

1. Introduction

Deep learning relies on backpropagation (BP) and automatic differentiation (AD), both are barriers in scenarios in hardware where gradients cannot be computed analytically.

- **Mono-Forward** [1] (MF) eliminates BP via layer-wise training with goodness-based loss functions, but still uses AD, related work shows comparable results to BP.
- **FFZero** [2] replaces AD with zeroth-order directional derivatives in a forward-forward setting, but uses prototype-based cosine similarity for goodness scores.

Aim: Adapt MF to use ZO optimization, achieving training that is entirely free of BP and AD, and evaluate its performance.

Layerwise Loss

$$\mathcal{G}_l \triangleq a_l \times M_l^\top$$

$$\mathcal{L}_l \triangleq -\sum_{c=1}^{m} y_c \log(\sigma(\mathcal{G}_{lc}))$$

Weight Updates (AD)

$$\mathbf{W}_l \leftarrow \mathbf{W}_l - \eta \left(\frac{\partial \mathcal{L}_l}{\partial \mathcal{G}_l} \cdot \frac{\partial \mathcal{G}_l}{\partial \mathbf{a}_l} \cdot \frac{\partial \mathbf{a}_l}{\partial \mathbf{z}_l} \cdot \frac{\partial \mathbf{z}_l}{\partial \mathbf{W}_l} \right)$$

$$\mathbf{M}_l \leftarrow \mathbf{M}_l - \eta \left(\frac{\partial \mathcal{L}_l}{\partial \mathbf{G}_l} \cdot \frac{\partial \mathbf{G}_l}{\partial \mathbf{M}_l} \right)$$

2. Directional Derivative-Based Approach

Zeroth-Order (ZO) gradient estimation [3,2] uses only two forward passes per direction. For direction $\mathbf{v}$:

$$\nabla_{\mathbf{v}} \mathcal{L}(\boldsymbol{\omega}) = \lim_{\varepsilon \to 0} \frac{\mathcal{L}(\boldsymbol{\omega} + \varepsilon \hat{\mathbf{v}}) - \mathcal{L}(\boldsymbol{\omega} - \varepsilon \hat{\mathbf{v}})}{2\varepsilon}.$$

$$\mathbb{E}_{\mathbf{v}}[n \nabla_{\mathbf{v}} \mathcal{L}(\boldsymbol{\omega}) \hat{\mathbf{v}}] = \frac{\partial \mathcal{L}(\boldsymbol{\omega})}{\partial \boldsymbol{\omega}}$$

Using this, for direction $\mathbf{v}$ layer weight and projection matrix updates are:

$$\boldsymbol{\omega}_l \leftarrow \boldsymbol{\omega}_l - \eta \, n_{\boldsymbol{\omega}} \, \nabla_{\boldsymbol{\omega}} \mathcal{L}_l(\boldsymbol{\omega}_l) \cdot \hat{\mathbf{v}}_{\boldsymbol{\omega}}$$

Where $\boldsymbol{\omega} \in \{W_l, M_l\}$

Problems:

- **DD** estimate is noisy, with a large number of parameters its noise is amplified even more!
- Mono-Forward also has an inherent problem of having lots of parameters in some architectures.

Solution:

- Split trainable parameters into **chunks** to reduce noise amplification
- Use **random projections** to reduce dimensionality of the channel-wise activations in architectures with large number of activations.

3. Research Questions

- **RQ1:** What is the accuracy cost of replacing AD with DD based zeroth-order methods in the Mono-Forward framework?
- **RQ2:** How does the number of perturbation directions **P** affect classification performance and model convergence?

4. Methodology

- **RQ1** is explored on MNIST, FashionMNIST, CIFAR10 and CIFAR100
- **RQ2** FashionMNIST with P ∈ {1, 2, 4, 8} directions
- Experiments ran on DelftBlue HPC

5. Results

RQ1: Accuracy Cost of Using Directional Derivatives. Cumulative-mode predictions are the primary metric, final-layer mode in parantheses

MLP

Dataset	BP	MF+AD	MF+DD (P=4)	Δ
MNIST	**0.973** ± 0.001	0.950 ± 0.001 (0.948 ± 0.001)	0.933 ± 0.002 (0.932 ± 0.000)	−0.017 (−0.015)
FashionMNIST	**0.879** ± 0.004	0.860 ± 0.003 (0.858 ± 0.004)	0.837 ± 0.002 (0.821 ± 0.000)	−0.023 (−0.037)
CIFAR-10	0.486 ± 0.004	**0.526** ± 0.004 (0.523 ± 0.003)	0.376 ± 0.001 (0.300 ± 0.001)	−0.150 (−0.222)
CIFAR-100	**0.208** ± 0.004	0.189 ± 0.003 (0.149 ± 0.002)	0.109 ± 0.004 (0.069 ± 0.002)	−0.080 (−0.080)

CNN

Dataset	BP	MF+AD	MF+DD (P=4)	Δ
MNIST	**0.989** ± 0.001	0.965 ± 0.001 (0.971 ± 0.001)	0.949 ± 0.002 (0.956 ± 0.003)	−0.016 (−0.015)
FashionMNIST	**0.911** ± 0.001	0.883 ± 0.004 (0.888 ± 0.004)	0.842 ± 0.004 (0.827 ± 0.003)	−0.041 (−0.061)
CIFAR-10	**0.731** ± 0.003	0.552 ± 0.007 (0.540 ± 0.006)	0.418 ± 0.009 (0.387 ± 0.004)	−0.134 (−0.153)
CIFAR-100	**0.430** ± 0.002	0.274 ± 0.003 (0.268 ± 0.002)	0.176 ± 0.005 (0.151 ± 0.004)	−0.098 (−0.117)

RQ2: Number of Perturbations vs. Accuracy and Effect on Convergence

Table 1: MF+DD Accuracy on FashionMNIST vs. P

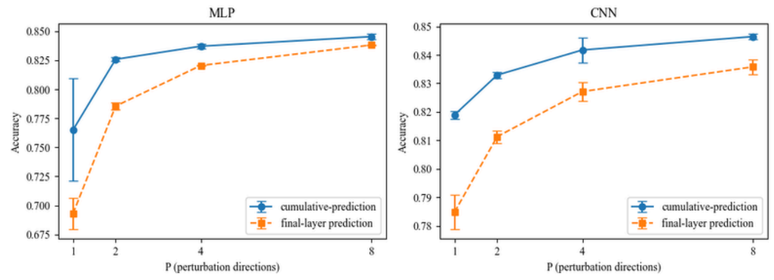
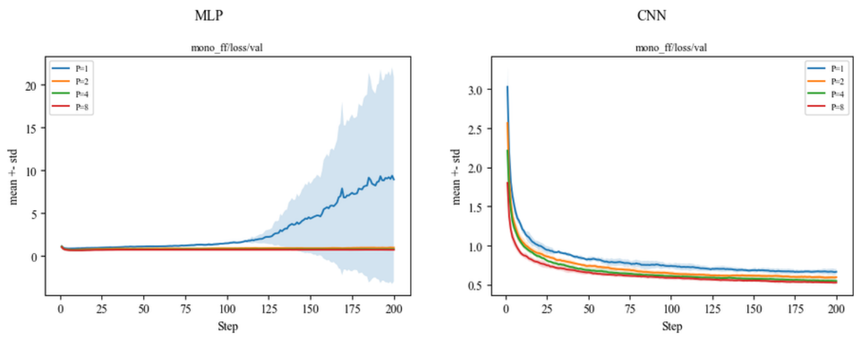


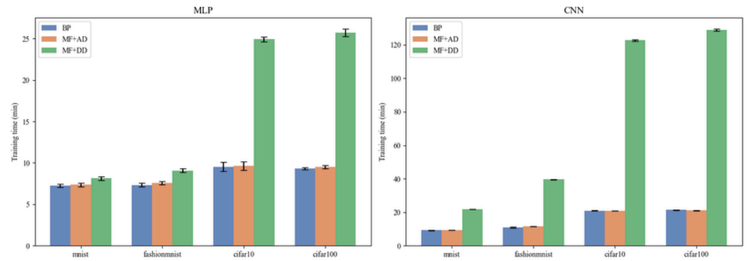
Table 2: Convergence Behavior with Different P values (Validation Loss)



6. Conclusions

- The cost of replacing AD with DD in MF is relatively **small** in simple datasets, while **complex datasets show larger gaps.**
- The number of perturbations stabilize training and increase accuracy, and reduce the noise as expected, but high values are performance bottlenecks.
- Training time is an issue with larger number of parameters and perturbation directions.

Table 3: Training Time of Different BP vs. MF+AD vs. VF+DD in Various Datasets



7. Future work

- Explore the effect of new hyperparameters introduced for DD approach, **chunk size** and **number of projection dimensions.**
- Experiments on actual physical hardware to determine feasibility in physical systems.
- Use optimizers such as Adam [4] , to see the effect of normalization mechanisms on noisy gradients.
- See the effect of allocating more perturbation directions to high loss layers, balancing gradient quality and computational cost.

7. References

[1] James Gong, Bruce Li, and Waleed Abdulla. Mono-forward: Revisiting forward-forward through objective-locality decomposition, 2026. URL https://arxiv.org/abs/2501.09238.

[2] Yaqi Guo, Fabian Braun, Bastiaan Ketelaar, Stephanie Tan, Richard Norte, and Siddhant Kumar. Local learning for stable backpropagation-free neural network training towards physical learning, 2026. URL https://arxiv.org/abs/2603.24790.

[3] Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O. Hero III, and Pramod K. Varshney. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. IEEE Signal Processing Magazine, 37(5):43–54, 2020. doi: 10.1109/MSP. 2020.3003837.

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.