RUB

TECHNISCHE UNIVERSITÄT DRESDEN

A variational framework for local learning with probabilistic latent representations

David Kappel[1], Khaleelulla Khan Nazeer[2], Cabrel Teguemne Fokam[1], Christian Mayr[2,3], Anand Subramoney[4]

Main findings

- **Backprop** introduces a **locking problem** - forward and backward phase **must wait** for each other (Jaderberg et al., 2016).
- The two phases rely on the **same weight matrices** to compute updates, known as the **weight transport problem** (Grossberg, 1987; Lillicrap et al., 2014).
- Locking and weight transport problems, make **parallelization inefficient**.
- We propose a new method to address these problems to **distribute a globally defined optimization algorithm across computing devices using only local updates**.
- Our approach is derived from **variational inference** that provides auxiliary local targets and communicates messages **forward and backward in parallel**.
- Within each block, conventional error backpropagation is performed locally - **Block Local Learning (BLL)**

A PROBABILISTIC FORMULATION OF BLOCK-LOCAL DISTRIBUTED LEARNING

Splitting a network into blocks k can be formalized by introducing latent variables $\mathbf{z}_k$

$$\alpha_k(\mathbf{z}_k) = p(\mathbf{z}_k|\mathbf{x}) = \mathbb{E}[p_k(\mathbf{z}_k|\mathbf{z}_{k-1})\alpha_{k-1}(\mathbf{z}_{k-1})] = f_k(\alpha_{k-1}, \boldsymbol{\theta}_k)$$

For every forward block k we introduce a **feedback block** (see Figure 1), such that

$$\rho_k(\mathbf{z}_k) = q_k(\mathbf{z}_k|\mathbf{x},\mathbf{y}) \quad \propto \quad p(\mathbf{z}_k|\mathbf{x})q(\mathbf{y}|\mathbf{z}_k) = \alpha_k(\mathbf{z}_k)\beta_k(\mathbf{z}_k)$$

Learning goal replaced by variational lower bound with block-local losses

$$\mathcal{F} = -\log p(\mathbf{y}|\mathbf{x}) + \frac{1}{N}\sum_{k=1}^{N}\mathcal{D}_{KL}(q_k|p_k) \quad \geq \mathcal{L} \quad = -\log p(\mathbf{y}|\mathbf{x})$$

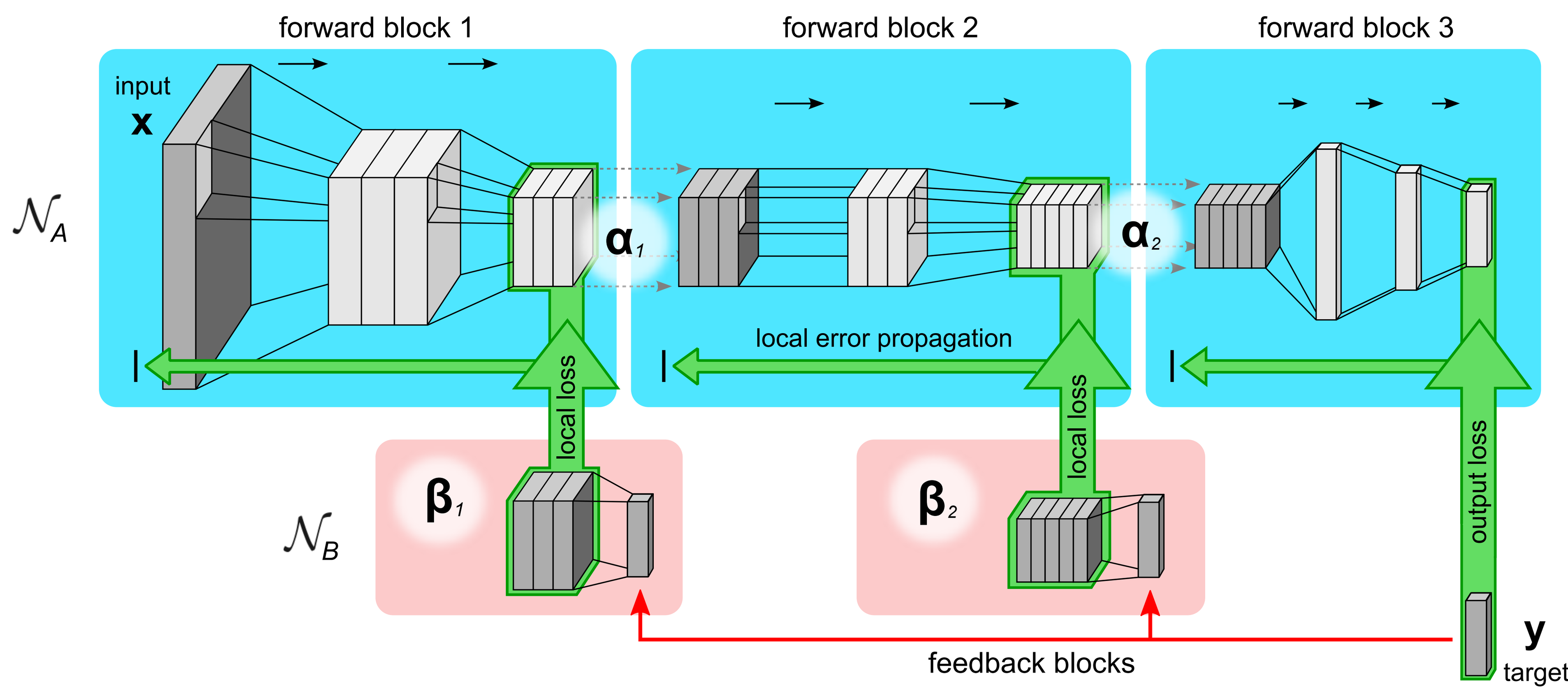

Figure 3: Block-local representations as learning signals. A deep neural network architecture is split into multiple blocks (forward blocks) and trained on an auxiliary local loss. Targets for local losses are provided by feedback blocks.

VARIATIONAL GREEDY BLOCK-LOCAL LEARNING

The BLL algorithm is shown in Figure 2. The two *for*-loops can be **interleaved and parallelized by pipelining the propagation** of data samples through the network.

```
for x,y in the training data set do
    α_0 ← x
    for 1 ≤ k ≤ N do
        β_k ← g_k(y, φ_k)                          ▷ Feedback
        α_k ← f_k(α_{k−1}, θ_k)                    ▷ Forward
        φ_k ← argmin_{φ_k} ℓ_k(p_k, β_k | α_{k−1})
        θ_k ← φ_k + η ∇_{θ_k} ℓ_k(p_k, β_k | α_{k−1})
```

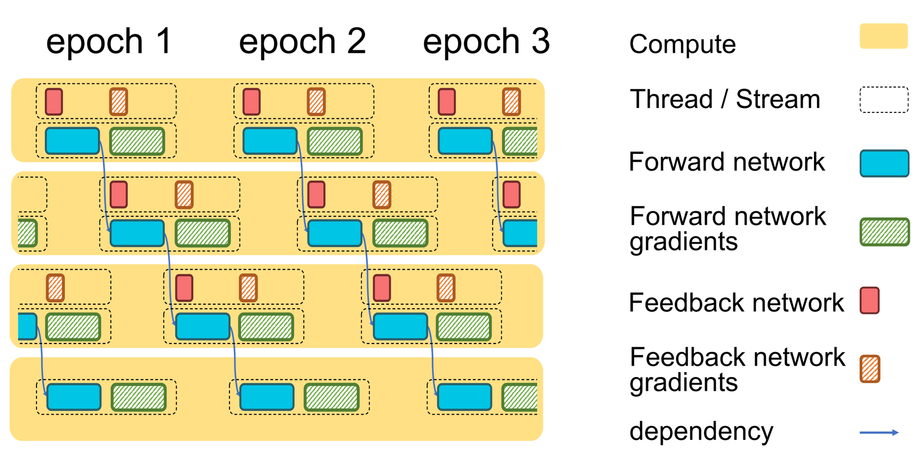

Figure 2: Left: Pseudo code of the BLL training algorithm. f_k and g_k are the transfer functions of forward and feedback blocks, respectively. The *for*-loops can be interleaved and run in parallel. **Right:** Timeline of execution for BLL.

ROYAL HOLLOWAY UNIVERSITY OF LONDON

BLOCK-LOCAL LEARNING OF VISION BENCHMARK TASKS

Evaluation (see Table 1):

- Fashion-Mnist and CIFAR-10
- ResNet18 and ResNet50 architectures
- Error Backpropagation (**BP**)
- Feedback Alignment (Lillicrap et al., 2014) (**FA**)
- Local learning using similarity matching loss (**Pred-Sim**) (Nøkland and Eidnes, 2019).
- Block Local Learning (**BLL**)

	Fashion-MNIST ResNet-18	Fashion-MNIST ResNet-50	CIFAR-10 ResNet-50
BP	92.7	93.4	94.0
FA	87.9	83.1	70.3
Pred-Sim	93.9	**94.3**	92.4
BLL	**94.2**	**94.3**	**92.6**

Table 1: Classification accuracy (% correct) on vision tasks. BP: end-to-end backprop, FA: Feedback Alignment, Sim Loss: Local learning with similarity matching loss (Nøkland and Eidnes, 2019), BLL: block local learning.

BLOCK-LOCAL TRANSFORMER ARCHITECTURE FOR SEQUENCE-TO-SEQUENCE LEARNING

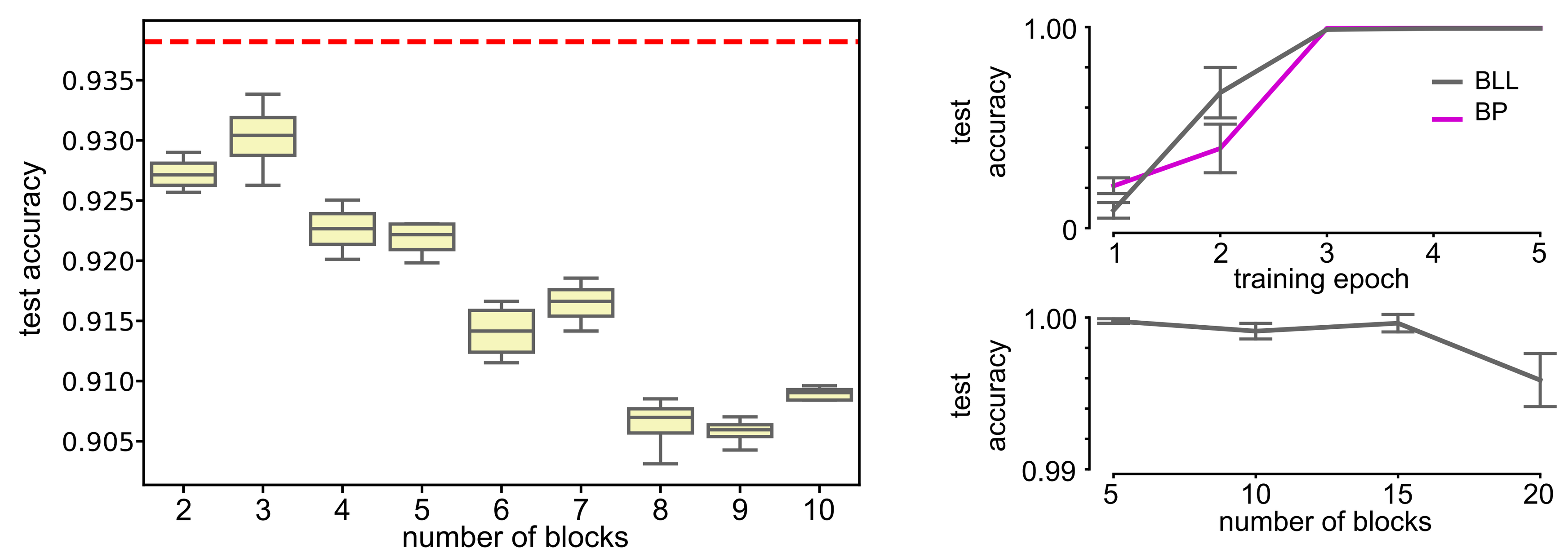

Figure 3: Scaling behavior of BLL. **A:** Test accuracy for different number of blocks for CIFAR-10 on ResNet-50. Dashed line shows BP baseline. **B:** Learning curves for S2S task. **C:** Test accuracy vs. number of blocks for S2S task. Error bars show standard deviations over independent runs.

Evaluation (see Figure 3):

- Transformer with 20 layers with a single attention head each.
- Block local losses after each layer and trained locally.
- task: S2S, random permutation of numbers 0..9 to be re-generated in reverse order.

Summary

- We address the problem of how can DNNs be efficiently distributed and horizontally scaled over many compute nodes.
- Our method is especially well suited for new energy efficient hardware for ML, such as edge devices.
- We use a probabilistic framework with block-local losses for training.
- Our initial results suggest that this new method performs on par or slightly better than previous related block-local learning approaches for small-scale tasks.
- The theoretical framework presented here is flexible and allows the introduction of complex, multi-layer feedback networks for which we show preliminary resutls on deep transformer networks.

[1]Institut für Neuroinformatik, Ruhr-Universität Bochum, Universitätsstr. 150 NB 3/32, 44801 Bochum
[2]Faculty of Electrical and Computer Engineering, and
[3]Centre for Tactile Internet with Human-in-the-Loop (CeTI), Technische Universität Dresden, Germany
[4]Royal Holloway, University of London, United Kingdom

{david.kappel,cabrel.teguemnefokam}@ini.rub.de
{khaleelulla.khan nazeer,christian.mayr}@tu-dresden.de
anand.subramoney@rhul.ac.uk

References

Jaderberg et al., 2016. Decoupled Neural Interfaces using Synthetic Gradients. http://arxiv.org/abs/1608.05343.

Stephen Grossberg, 1987. Competitive learning: From interactive activation to adaptive resonance. Cog. Sci. 11.

Lillicrap et al., 2014. Random feedback weights support learning in deep neural networks. http://arxiv.org/abs/1411.0247.

Nøkland and Eidnes, 2019. Training neural networks with local error signals. ICML. https://arxiv.org/abs/1901.06656

Acknowledgments: DK and CTF are funded by the German Federal Ministry of Education and Research (**BMBF**) within the **project EVENTS** (**16ME0733**). KKN is funded by the German Federal Ministry of Education and Research (**BMBF**) within the **KI-ASIC project** (**16ES0996**). We acknowledge the use of **Fenix Infrastructure resources**, which are partially funded from the European Union's Horizon 2020 research and innovation programme through the ICEI project under the **grant agreement No. 800858**. CM receives funding from the German Research Foundation (**DFG**, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy – EXC 2050/1 – **Project ID 390696704** – Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI) of TU Dresden.