UNLOCKING DEEP LEARNING: A BP-FREE APPROACH FOR PARALLEL BLOCK-WISE TRAINING OF NEURAL NETWORKS

*Anzhe Cheng**   *Zhenkun Wang**   *Chenzhong Yin**   *Mingxi Cheng**
*Heng Ping**   *Xiongye Xiao*   *Shahin Nazarian*   *Paul Bogdan*

University of Southern California, Los Angeles, CA, USA

ABSTRACT

Backpropagation (BP) has been a successful optimization technique for deep learning models. However, its limitations, such as backward- and update-locking, and its biological implausibility, hinder the concurrent updating of layers and do not mimic the local learning processes observed in the human brain. To address these issues, recent research has suggested using local error signals to asynchronously train network blocks. However, this approach often involves extensive trial-and-error iterations to determine the best configuration for local training. This includes decisions on how to decouple network blocks and which auxiliary networks to use for each block. In our work, we introduce a novel BP-free approach: a block-wise BP-free (BWBPF) neural network that leverages local error signals to optimize distinct sub-neural networks separately, where the global loss is only responsible for updating the output layer. The local error signals used in the BP-free model can be computed in parallel, enabling a potential speed-up in the weight update process through parallel implementation. Our experimental results consistently show that this approach can identify transferable decoupled architectures for VGG and ResNet variations, outperforming models trained with end-to-end backpropagation and other state-of-the-art block-wise learning techniques on datasets such as CIFAR-10 and Tiny-ImageNet. The code is released at https://github.com/Belis0811/BWBPF.

Index Terms— Local loss, block-wise learning, computer vision

1. INTRODUCTION

Based on the remarkable success of gradient descent training techniques [1], the Backpropagation (BP) has emerged as a widely adopted core learning algorithm in most deep learning networks. This algorithm efficiently calculates weight parameter gradients by reverse-propagating the error signal from the loss function to each layer. However, BP faces certain limitations, characterized by update locking and backward locking [2], where update locking necessitates the completion of a forward pass before any weight updates can occur, while backward locking requires gradient computation in upper layers to precede that in lower layers.

An intriguing category of alternative proposals is grounded in contrastive learning within energy-based models [3, 4]. Rather than propagating errors layer by layer through random feedback connections, Nøkland [5] and Neftci *et al.*'s networks [6] employ a fixed random projection of the top layer error as the error signal in deep layers, which enables a unified error signal shared across all layers. However, this approach results in delays and high memory demands during weight updates. A complete forward pass through the network is necessary before the error signal becomes accessible, leading to deep layers maintaining states throughout, consuming substantial computational resources.

To address these challenges, researchers have introduced a solution in the form of greedy block-wise learning approaches [7, 8]. These methods partition the network into distinct blocks, each comprising consecutive layers connected to an auxiliary network. The auxiliary network computes local loss, optimizing block weights by propagating error signals. This scheme enables independent parameter updates in individual blocks during concurrent forward pass computations. Recently, Gunhee *et al.* [9] proposed SEDONA, an automatic search method for efficient exploration of decoupled neural architectures using greedy block-wise learning. SEDONA improves validation loss optimization by organizing layers into blocks and selecting suitable auxiliary networks. However, in deeper networks with more blocks, SEDONA's classification performance shows marginal improvement or remains similar to standard BP method.

To fully utilize the potential of the block-wise local loss approach, we present a novel BP-free learning approach based on the block-wise BP-free (BWBPF) learning method. Leveraging the original architecture's blocks as a foundation, we partition the network into several subnetworks connected via our auxiliary network. These subnetworks are connected to independent dense layers, which are responsible for calculating local losses. These local loss terms are then utilized to update the parameters of each individual block. Through experiments on two image classification datasets, CIFAR-10 and Tiny-ImageNet, our method outperforms the original BP algorithm in ResNet [10] and VGG [11] networks. Fur-

---

*All authors contributed equally to this work

thermore, we compare our results with three greedy learning algorithms, DGL [7], PredSim [12], and the SEDONA method [9]. Our network consistently outperforms networks trained using those algorithms. Notably, unlike most of the aforementioned methods, our approach doesn't necessitate additional memory blocks to generate an error signal.

2. METHOD

The BWBPF learning approach draws inspiration from biological networks such as the human brain, where synaptic weight updates can occur through local learning, independent of the activities of neurons in other brain regions [13–17]. Partly for this reason, local learning has been identified as an effective means to reduce memory usage during training and to facilitate parallelism in deep learning architectures.

2.1. Block-wising learning in convolutional blocks

In the context of a convolutional neural network (CNN) comprised of multiple layers, the concept of "blocks" emerges as a convenient way to organize and group one or more convolutional layers together. To simplify this concept further, when dealing with a neural network featuring multiple convolutional layers, we can naturally group a set of these layers together into what we refer to as a "block." This grouping approach is widely employed in well-known architectures like VGG [11], ResNet [10], and Inception [18], where these blocks play a pivotal role in shaping the network's structure and behavior.

Following each convolutional block in our model, we incorporate two essential components: a global average pooling layer and a dense layer responsible for computing local loss. In this work, we exclusively employ cross-entropy loss functions. These auxiliary components within the network, as depicted in Fig. 1 (b), hold the crucial responsibility of updating the layer weights within each block. Importantly, they carry out this weight-updating task while preserving the overall structural integrity of the network, ensuring that the underlying architecture remains intact. This architectural choice enables our network to consistently deliver superior performance when compared to networks trained using alternative algorithms. It is noteworthy that our approach achieves this performance without the need for additional memory blocks to generate an error signal, a departure from many of the existing methodologies in the field.

2.2. BP-free

To overcome the drawbacks of BP, particularly the issue of backward locking, we propose the BWBPF learning algorithm which eliminates BP for the global prediction loss and instead computes the local prediction loss. This modification implies that the global prediction loss is solely responsible for optimizing the output layer, denoted as $\mathcal{L}_g^o$ (where o stands for output). In simpler terms, this prediction loss acts as a local loss function exclusively for updating the weights of the output layer (as depicted in Fig. 1 (c)). The total loss function for

the BP-free algorithm is defined as follows:

$$\mathcal{L} = \lambda_1 \mathcal{L}_g^o + \lambda_2 \sum_{l=1}^{K} \mathcal{L}_l^\delta \tag{1}$$

where $\mathcal{L}_l^\delta$ represents the local loss, with l denoting the number of blocks. The local loss is computed between the true label and the local output vector ($\vec{y_\delta}$). By removing the BP of the global prediction loss to the hidden layers, the weight updates for the convolutional layers in our model are exclusively influenced by the local prediction loss, as illustrated in Figure 1 (c).

To enable layers in convolutional blocks to generate valid predictions, we apply the same activation function used in the output layer for each block. For instance, in the case of a CNN designed for N-class classification, we typically include N output neurons in the output layer and apply the softmax function. In this work, we equally group the layers into K different blocks and embed our approach on VGG-19, ResNet50, ResNet101 and ResNet152 for validating our model's performance by spilt the original architecture into 4, 8, 12, and 16 blocks. We conducted model training for 200 epochs with CIFAR-10 and 300 epochs with Tiny-ImageNet, using a batch size of 32. We initially set the learning rate to 0.1 and gradually reduced it to 0.0001. For the ResNet-50/101/152 and VGG-19 architectures, we applied a weight decay of 0.0001 and a momentum of 0.9. Additionally, we utilized weight initialization without employing dropout.

3. EXPERIMENTS

To validate our approach and make a meaningful comparison with state-of-the-art block-wise learning methods, we employ our approach and baseline methods on both VGG-19 [11] and ResNet-50/101/152 [10] architectures. To ensure a fair comparison, all methods are trained and tested on identical datasets, which include CIFAR-10 and Tiny ImageNet. Furthermore, we maintain uniformity in the training strategy and hyperparameter settings across all approaches.

Table 1: Error rates(%) on CIFAR-10 (a) and Tiny-ImageNet (b) with different methods with 4 outputs

Architecture	*BP	*PredSim	*DGL	*SEDONA	BWBPF
VGG-19	12.31	13.87	12.19	11.58	**6.52**
ResNet50	7.99	8.93	8.27	7.53	**4.85**
ResNet101	7.14	7.93	8.30	6.59	**4.53**
ResNet152	6.35	7.41	6.39	6.13	**4.48**

*Results are from SEDONA.

(a) CIFAR-10

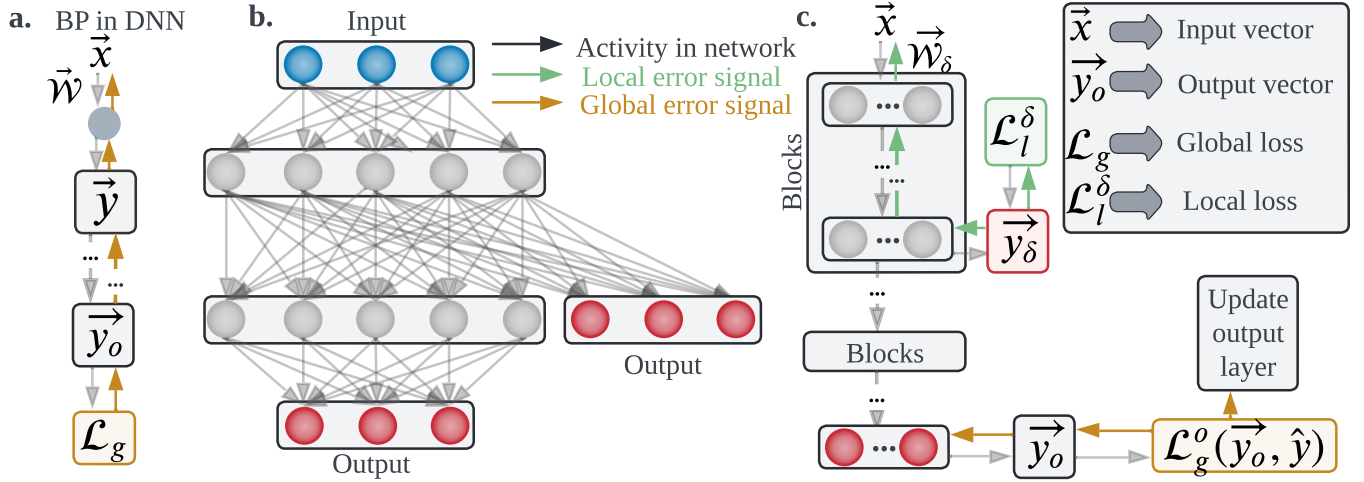| Architecture	*BP	*PredSim	*DGL	*SEDONA	BWBPF
VGG-19	47.11	55.30	48.70	43.44	**40.09**
ResNet50	46.54	55.22	46.04	45.60	**35.83**
ResNet101	44.50	46.80	46.20	40.88	**35.28**
ResNet152	39.18	48.24	42.36	35.90	**35.01**

*Results are from SEDONA.

(b) Tiny-ImageNet

Fig. 1: **Weight updates of BWBPF. a.** BP in classic neural network training. Global prediction loss is back-propagated through layers. **b.** The dense layer is attached below the subnetwork for computing the local loss. **c.** BP-free weight update of a block of layers.
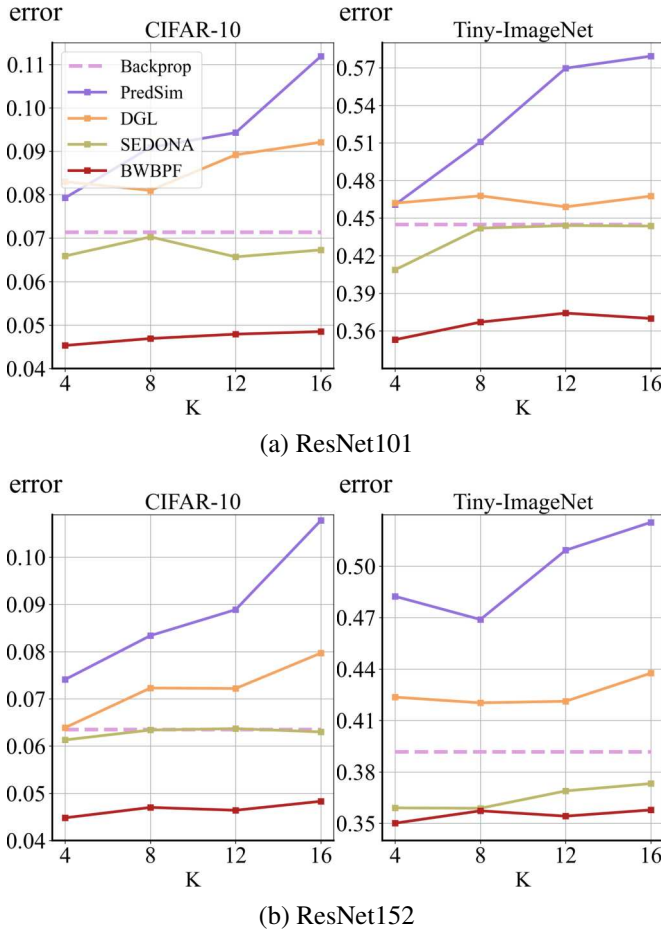

(a) ResNet101

(b) ResNet152

Fig. 2: Classification error (1-accuracy) curves of (a) ResNet-101 and (b) -152 with increasing K on CIFAR-10 and Tiny-ImageNet.

3.1. Datasets

The CIFAR-10 is obtained from the TensorFlow datasets [19]. CIFAR-10 [20] consists of 60,000 images, each of size 32×32. Tiny ImageNet [21] consists of a dataset of $100,000$ images distributed across 200 classes, with 500 images per class for training, and an additional set of $10,000$ images for testing. All images in the dataset are resized to 64×64 pixels.

3.2. Baselines models

In this paper, we choose three state-of-the-art block-wise learning algorithms which use BP-free optimization, which are DGL [7], PredSim [12], and SEDONA [9]: (1) DGL uses greedy block-wise learning method to train their network by dividing it into K subnetworks. The auxiliary network they use denoted as MLP-SR-aux, which consists of one pooling layer with three point-wise convolutions followed by an average pooling layer and a 3-layer MLP; (2) Pred-Sim combines two local losses, similarity matching loss and cross-entropy loss, which require two different auxiliary networks, an average pooling layer with fully connected layer and a convolutional layer, respectively; (3) SEDONA generates four auxiliary candidates, encompassing a sequence of components: a point-wise convolutional layer, a depth-wise convolutional layer, a specific number of inverted residual blocks, and ultimately concluding with a point-wise convolutional layer. Subsequently, each candidate undergoes further processing, involving the application of an average pooling layer and the integration of a fully connected layer.
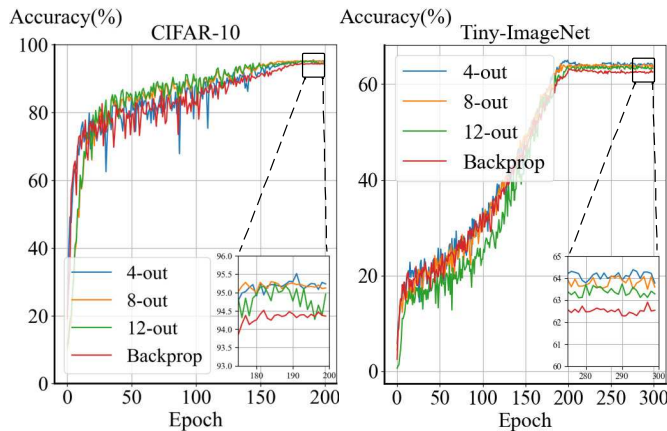
3.3. Results on CIFAR-10 and Tiny-ImageNet

Our primary classification error rates (1-accuracy) for both CIFAR-10 and Tiny-ImageNet datasets are summarized in Table 1 (BP results are also shown in Table 1 for reference). For the CIFAR-10 dataset, noted that two state-of-the-art

4237

block-wise learning methods, PredSim and DGL, almost underperform traditional BP in all cases. In contrast, the network trained with the SEDONA method shows a better performance than BP. However, our model has a more significant improvement than the SEDONA method. Our image classification accuracy is about **5%** higher on VGG-19 than on SEDONA, and about **2%** higher on all three ResNet networks.
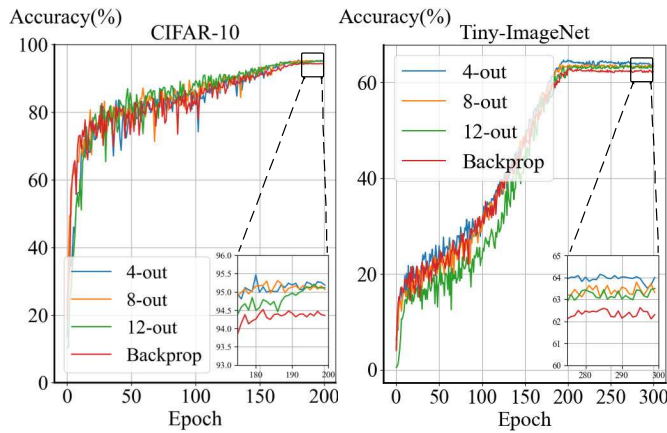
(a) ResNet101

(b) ResNet152

Fig. 3: Classification accuracy curves of (a) ResNet-101 and (b) -152 with increasing K and end-to-end BP on CIFAR-10 and Tiny-ImageNet.

Similarly, our method not only outperforms BP but also yields better performances to SEDONA on Tiny-ImageNet. As indicated in Table 1b, our approach's accuracy even surpassed SEDONA by over **10%** on ResNet50. For the remaining ResNet architectures, the optimization gain from our method gradually diminished as the model complexity increased with additional layers. Nevertheless, our method consistently outperforms SEDONA across all configurations. We achieved a **5%** improvement in accuracy compared to SEDONA with ResNet101, and we also observed a slight

performance boost over SEDONA with ResNet152. These results affirm that our method presents a superior option for both large and small datasets compared to conventional BP and other block-wise learning models.

To investigate the impact of the number of blocks (K) on model performance, Figure 2 presents results from experiments conducted using four different learning approaches: DGL, PredSim, SEDONA, and our BWBPF learning, each with varying values of K. Upon analyzing the baseline methods, it is evident that both DGL and PredSim consistently underperform compared to the standard BP method across all values of K, particularly for $K \geq 12$. On the other hand, SEDONA exhibits superior performance to BP, particularly for lower values of K. However, as K increases, SEDONA's performance becomes comparable to that of BP, especially when validating on CIFAR-10 and Tiny-ImageNet datasets using the ResNet 101 model. In our model, we observe that increasing the value of K leads to an increase in the error rate. Nevertheless, our model consistently outperforms the classic BP algorithm and all the baseline models in terms of overall model performance, even as K increases.

4. CONCLUSION

To address the challenges posed by standard BP, we introduce a novel approach called the BWBPF learning algorithm, which operates without relying on BP. BWBPF, which divides the network into subnetworks connected to auxiliary networks, enables the computation of local losses for individual blocks, resulting in enhanced convergence, reduced gradient issues, better generalization, and increased model robustness. Through comprehensive experiments on image classification tasks using popular architectures such as ResNet and VGG, we showcase the superiority of our approach over traditional BP and existing state-of-the-art greedy learning methods. BWBPF's adaptability to various network architectures and datasets indicates its potential for broader applications in the field of deep learning. As we continue to explore alternative learning mechanisms inspired by biology, the proposed approach offers a promising avenue for improving the efficiency and effectiveness of training CNN architectures.

5. ACKNOWLEDGEMENT

6. REFERENCES

[1] Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan, "Estimating training data influence by tracing gradient descent," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19920–19930, 2020.

[2] Max Jaderberg, Wojciech Marian Czarnecki, Simon Osindero, Oriol Vinyals, Alex Graves, David Silver, and Koray Kavukcuoglu, "Decoupled neural interfaces using synthetic gradients," in *International conference on machine learning*. PMLR, 2017, pp. 1627–1635.

[3] Yoshua Bengio and Asja Fischer, "Early inference in energy-based models approximates back-propagation," *arXiv preprint arXiv:1510.02777*, 2015.

[4] Benjamin Scellier and Yoshua Bengio, "Equilibrium propagation: Bridging the gap between energy-based models and backpropagation," *Frontiers in computational neuroscience*, vol. 11, pp. 24, 2017.

[5] Arild Nøkland, "Direct feedback alignment provides learning in deep neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[6] Emre O Neftci, Charles Augustine, Somnath Paul, and Georgios Detorakis, "Event-driven random backpropagation: Enabling neuromorphic deep learning machines," *Frontiers in neuroscience*, vol. 11, pp. 324, 2017.

[7] Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon, "Decoupled greedy learning of cnns," in *International Conference on Machine Learning*. PMLR, 2020, pp. 736–745.

[8] Sindy Löwe, Peter O'Connor, and Bastiaan Veeling, "Putting an end to end-to-end: Gradient-isolated learning of representations," *Advances in neural information processing systems*, vol. 32, 2019.

[9] Myeongjang Pyeon, Jihwan Moon, Taeyoung Hahn, and Gunhee Kim, "Sedona: Search for decoupled neural networks toward greedy block-wise learning," in *International Conference on Learning Representations*, 2020.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[11] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[12] Arild Nøkland and Lars Hiller Eidnes, "Training neural networks with local error signals," in *International conference on machine learning*. PMLR, 2019, pp. 4839–4850.

[13] Natalia Caporale and Yang Dan, "Spike timing–dependent plasticity: a hebbian learning rule," *Annu. Rev. Neurosci.*, vol. 31, pp. 25–46, 2008.

[14] Chenzhong Yin, Phoebe Imms, Mingxi Cheng, Anar Amgalan, Nahian F Chowdhury, Roy J Massett, Nikhil N Chaudhari, Xinghe Chen, Paul M Thompson, Paul Bogdan, et al., "Anatomically interpretable deep learning of brain age captures domain-specific cognitive impairment," *Proceedings of the National Academy of Sciences*, vol. 120, no. 2, pp. e2214634120, 2023.

[15] Phoebe Imms, Jaron Kawamura, Chenzhong Yin, Anar Amgalan, Nahian F Chowdhury, Nikhil N Chaudhari, and Andrei Irimia, "Early identification of alzheimer's risk using anatomically interpretable estimation of brain age," in *Alzheimer's Association International Conference*. ALZ, 2023.

[16] Mikael Lundqvist, Scott L Brincat, Jonas Rose, Melissa R Warden, Timothy J Buschman, Earl K Miller, and Pawel Herman, "Working memory control dynamics follow principles of spatial computing," *Nature Communications*, vol. 14, no. 1, pp. 1429, 2023.

[17] Chenzhong Yin, Xiongye Xiao, Valeriu Balaban, Mikhail E Kandel, Young Jae Lee, Gabriel Popescu, and Paul Bogdan, "Network science characteristics of brain-derived neuronal cultures deciphered from quantitative phase imaging data," *Scientific reports*, vol. 10, no. 1, pp. 15078, 2020.

[18] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[19] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., "Tensorflow: a system for large-scale machine learning.," in *Osdi*. Savannah, GA, USA, 2016, vol. 16, pp. 265–283.

[20] Alex Krizhevsky, Geoffrey Hinton, et al., "Learning multiple layers of features from tiny images," 2009.

[21] Ya Le and Xuan Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, pp. 3, 2015.