A SURVEY OF BACKPROPAGATION-FREE TRAINING FOR LLMS*

Hanzi Mei, Dongqi Cai, Yaozong Wu, Shangguang Wang, Mengwei Xu
Beijing University of Posts and Telecommunications (BUPT)
Contact: cdq@bupt.edu.cn
Website: https:github.com/UbiquitousLearning/Backpropagation_Free_Training_Survey

ABSTRACT

Large language models (LLMs) have achieved remarkable performance in various downstreaming tasks. However, the training of LLMs is computationally expensive and requires a large amount of memory. To address this issue, backpropagation-free (BP-free) training has been proposed as a promising approach to reduce the computational and memory costs of training LLMs. In this survey, we provide a comprehensive overview of BP-free training for LLMs from the perspective of mainstream BP-free training methods and their optimizations for LLMs. The goal of this survey is to provide a comprehensive understanding of BP-free training for LLMs and to inspire future research in this area.

1 Introduction

In recent years, the field of machine learning has witnessed a remarkable evolution, predominantly driven by the advent and proliferation of large language models (LLMs) such as GPTs. These models have demonstrated unparalleled proficiency in a wide range of generic machine learning tasks, significantly advancing the capabilities of artificial intelligence. Traditionally, most LLMs, if not all, have been trained using a forward-backward paradigm. This paradigm predominantly relies on backwardpropagation (BP) to compute gradients, which are essential for updating the model weights. Despite its effectiveness, BP-based methods encounter several limitations as the scale of models increases such as hugh resource cost, huge memory footprint, and incompatibility with device accelerators (§2.1).

In light of these challenges, there has been a growing interest in exploring backpropagation-free (BP-free) training methods (§2.2). These alternative approaches primarily focus on various forms of perturbation - including model perturbation, input perturbation, and approaches that do not involve perturbation - to achieve learning without the need for traditional backpropagation (§3).

While the majority of these methods were not initially designed for training LLMs, recent advancements and innovative proposals have begun to explore and refine these techniques for use in large-scale model training (§4). These developments suggest a promising direction for overcoming the current limitations of BP-based methods, potentially democratizing access to the BP-free training for LLMs.

This survey aims to provide a comprehensive overview of the advancements and methodologies in BP-free training of LLMs, offering insights into the future trajectory of scalable and sustainable machine learning practices.

2 Background

2.1 Large Language Model

Large language models (LLMs) have marked a paradigm shift in the field of artificial intelligence, fundamentally transforming our approach to a myriad of complex tasks. The inception of this transformative era can be traced back to the development of the Transformer architecture by Google. This architecture, distinguished by its self-attention mechanisms, allows for more effective processing of sequential data compared to its predecessors. Transformers have set a new standard in the field, particularly in tasks involving natural language processing and computer vision.

*This work is still in progress.

At the core of the Transformer architecture is its ability to handle dependencies in data irrespective of their distance in the sequence. This capability is primarily facilitated by the self-attention mechanism, which computes the response at a position in a sequence by attending to all positions and weighting them according to their relevance. This process results in a model that is highly adept at capturing complex relationships within data, making it ideal for tasks such as language translation, content generation, and image recognition.

The common workflow for training these LLMs involves a forward-backward process centered around backpropagation (BP). Initially, the model processes input data in a forward pass, generating predictions. The predictions are then compared against the actual outcomes, and the discrepancy (loss) is measured. During the backward pass, gradients are computed by backpropagating this loss through the network, allowing for the adjustment of weights in the model. This iterative process of forward prediction and backward adjustment is crucial for the model to learn from data effectively.

However, this training approach brings with it a set of significant challenges:

Huge Resource Cost. The computational resources required for BP-based training of LLMs are enormous. The energy demands for training such models often exceed the annual energy usage of countries like New Zealand and Austria, posing severe environmental and economic implications [47]. This not only burdens large enterprises capable of undertaking such training but also creates an insurmountable barrier for smaller entities and individuals.

Huge Memory Footprint. BP-based training also demands substantial memory resources. In addition to handling billions of model weights, the training process necessitates storing numerous intermediate activations for the backward pass. This vast memory requirement further complicates the training process, limiting the scope of training to systems with extensive memory capabilities.

Incompatibility with Device Accelerators. Lastly, the training of LLMs is often incompatible with many common devices, like mobile phones, due to their incapacity to support such extensive computational and memory requirements. This limitation significantly hinders the democratization and widespread adoption of large-scale model training, confining it to entities equipped with advanced computing infrastructures.

These challenges underscore the need for alternative training methodologies that could alleviate the resource and memory constraints while widening the accessibility of LLM training across diverse platforms and devices.

2.2 Backpropagation-free Training

The quest for deriving optimization directions for models without relying on backpropagation has been a subject of considerable research interest. Early backpropagation-free (BP-free)approaches rooted in direct search, such as coordinate search [13] and pattern search [42]. Additionally, model-based strategies like model-based descent [4] and trust region [7] methods have also been explored. Evolutionary strategies, inspired by the mechanisms of natural evolution, iteratively refine a set of solutions, aiming to discover the optimal or near-optimal solution to a given problem. This iterative process emulates the evolutionary dynamics observed in nature, facilitating the exploration of solution spaces to achieve desired optimization goals.

In recent years, perturbation-based forward propagation methods have garnered significant attention, notably the forward gradient and zero-order optimization techniques. Both methods involve steering the function's descent along a randomly selected direction. This can be traced back to the earliest literature [31] and [38]. To implement these methods, a random direction v is chosen, and optimization is performed along this direction with a step size equivalent to the directional derivative of the objective function in v:

$$w = w - l_r * (g \cdot v)v \tag{1}$$

Here, $(g \cdot v)v$ represents an unbiased estimate of the true gradient. Recently, this method has garnered significant attention within the research community. In their seminal work, Baydin et al. [2] introduced the forward gradient technique. Leveraging automatic differentiation of forward mode, they effectively compute the directional derivative $(g \cdot v)v$ of the objective function with respect to a specified direction. Subsequently, they define $(g \cdot v)v$ as the forward gradient, thus supplanting the true gradient for gradient descent optimization.

Akin to forward gradients, zeroth-order optimization eschews automatic differentiation in favor of employing the finite difference method to compute directional derivatives. Subsequently, akin to the forward gradient method, perturbations and their corresponding directional derivatives are leveraged for optimization.

Benefits of BP-free Training The mentioned BP-free propagation algorithm does not necessitate the storage of activation values during computation, thereby circumventing the substantial memory overhead inherent in backpropa-

gation. Furthermore, the zero-order optimization method operates independently of modern deep learning software's automatic differentiation function, enhancing compatibility with inference accelerators deployed across diverse devices.

3 Backpropagation-free Training Methods

We introduce three mainstream Backpropagation-free (BP-free) training methods and their optimizations in this section. Most of the forward training methods are based on model/input perturbations and then optimized based on the perturbed function values. The detailed taxonomy is shown in Fig. 1.

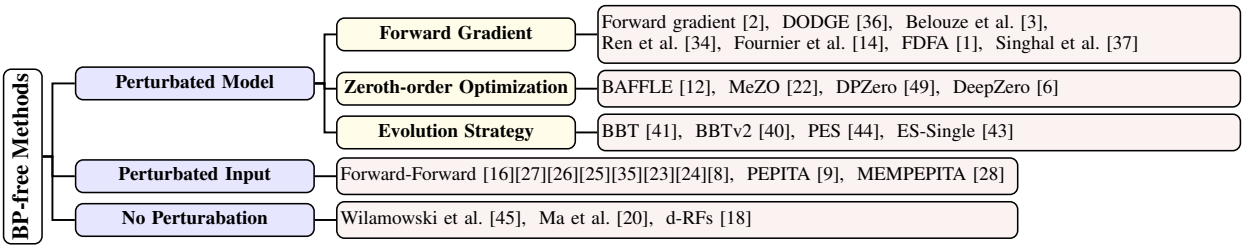

Figure 1: A taxonomy of BP-free training methods.

3.1 Perturbated Model

3.1.1 Forward Gradient

The forward gradient method, as proposed by [2], has recently garnered widespread attention. They advocate for employing forward mode automatic differentiation to compute directional derivatives of the objective function in random directions, generating forward gradients as a replacement for traditional backpropagation gradients, and facilitating gradient descent. Experimental results demonstrate that this approach significantly reduces computational complexity and enhances training speed. Consistent with the principles of the forward gradient method, Silver et al. [36] explore its application in optimizing recursive functions. They compute directional derivatives along candidate directions and employ them for parameter updates. Additionally, they scrutinize various methods for obtaining candidate directions, including random selection and approximation of actual gradients. In contrast, Belouze [3] argue that high-dimensional problems pose a significant challenge for forward gradients. In cases where the dimensionality of optimization parameters is exceedingly large, forward gradients exhibit pronounced variance, resulting in substantial deviation from the true gradient.

To diminish the variance of forward gradients, Ren et al.[34] suggests applying perturbations to activations instead of weights. Additionally, they advocate for the incorporation of a substantial number of local losses to constrain the number of learnable dimensions. This approach ultimately enhances the scalability of forward gradients. In a similar vein, Fournier et al. [14] proposes a strategy wherein gradient guessing is markedly biased towards more promising directions. Specifically, leveraging local losses from small auxiliary networks to ascertain gradient directions significantly diminishes random noise in forward gradient methods. Addressing the challenge of high squared bias in large-scale deep neural networks, Bacho et al. [1] puts forth the forward direct feedback alignment algorithm. This method employs activation perturbation forward gradients as direct feedback connections and integrates momentum methods, ultimately achieving diminished variance. Singhal et al. [37] explores how to more accurately guess the true gradient direction, which can improve gradient free optimization algorithms based on directional derivatives. Analysis and experiments have shown that gradient guessing with higher cosine similarity to the real gradient can be generated based on network architecture and incoming features.

3.1.2 Zeroth-order Optimization

Zero-order optimization commonly employs finite differences to approximate gradients. Feng et al. [12] introduces BAFFLE, a federated learning system devoid of backpropagation. BAFFLE utilizes zero-order optimization to estimate gradients by substituting backward processes with multiple forward processes. By synchronizing perturbations using random seeds, BAFFLE only uploads loss differences, rendering it adaptable to upload bandwidth constraints and well-suited for trusted execution environments Malladi et al. [22] presents MeZO, a memory-efficient zero-order optimizer. MeZO employs fixed random seeds for in-place perturbations, resulting in memory consumption comparable to inference during the optimization process. Combining zero-order optimization with differential privacy, Zhang

et al. [49] proposes the DPZero algorithm. Leveraging estimated gradients, DPZero exhibits convergence speed independent of dimensionality. Chen et al. [6] leverages zero-order optimization for model optimization, enhances gradient sparsity through pruning, and integrates feature reuse and forward parallelization. Consequently, they achieve state-of-the-art accuracy on a ResNet-20 model trained on CIFAR-10.

3.1.3 Evolution Strategy

Evolutionary strategy, a swarm intelligence algorithm, has recently found applications in black box optimization of large models and unrolled computation graphs.

In black box scenarios such as API inference calls, where the model gradient is unavailable, Sun et al. [41] proposes a hybrid approach combining prompt-based learning and evolutionary algorithms. This method optimizes continuous prompts preceding input text, surpassing the effectiveness of GPT-3's in-context learning. BBTv2[40] is an enhanced iteration of BBT. Unlike its predecessor, BBTv2 employs a divide-and-conquer algorithm to optimize prompts at each layer alternately. Additionally, it utilizes a model-specific normal distribution for random prompt projection, reducing the number of tunable parameters while maintaining performance.

In scenarios involving unrolled computation graphs, Vicol et al. [44] presents the Persistent Evolution Strategy (PES). PES conducts evolutionary strategy-based update steps on a series of truncated unfolding sequences, mitigating bias introduced by truncation through the accumulation of correction terms across the entire unfolding sequence. Experimental results demonstrate the broad applicability of this method across multiple tasks. Furthermore, Vicol et al. [43] proposes ES-Single as an unbiased gradient estimator for unrolled computational graphs, akin to PES. ES-Single samples parameter perturbations once at the outset of each inner problem and employs the same perturbations across each partial unroll. Notably, it achieves lower variance than PES in practice.

3.2 Perturbated Input

Some studies focus on how to perturb the input data rather than the model. Specifically, these approaches conduct forward propagation not only with the original input data but also with its modified variants. The direction for updating model parameters is then determined based on the responses of the hidden layers to these two types of forward propagation.

The Forward-Forward(FF) Algorithm [23] propses a training mechanism that employs two operationally identical forward passes, differentiated only by the input data and their optimization objectives. The first pass processes positive (i.e., real) data aiming to maximize layer-wise 'goodness', whereas the second pass utilizes negative data with the goal of minimizing the same metric. The FF training paradigm has been demonstrated to be effective across a multitude of network architectures. Initially applied to fully connected networks, its efficacy has since been established in various other structures, including Graph Neural Networks [27], Recurrent Neural Networks [26], Spiking Neural Networks [25], and Convolutional Neural Networks [35]. Furthermore, it is inherently more suitable for low-power analog hardware compared to the backpropagation (BP) algorithm. The FF algorithm, adapted for resource-constrained environments like wave-based physical platforms [23][24] and Micro-Controller Units (MCUs)[8], facilitates neural network training beyond the von Neumann architecture's scope.

Additionally, the Present the Error to Perturb the Input To modulate Activity (PEPITA) method [9] considers forward training as a credit assignment problem. This method initially conducts a standard forward propagation, followed by modulating the input data based on errors for an additional forward propagation. Neuronal updates are then calculated based on the differences between these two responses. Building upon this, the MEMPEPITA algorithm [28] proposes a variant involving three forward propagations, where activations and errors from the first forward pass are not stored, but instead, a third standard forward propagation is executed during the modulated forward pass. This approach aims for memory efficiency, albeit introducing additional computational overhead.

3.3 No perturbation

Wilamowski et al. [45] introduce a BP-free training methodology that obviates the need for perturbation. This technique extends the concept of signal gains, typically observed between neurons and outputs, to encompass interneuronal interactions. The computation of the signal gain matrix directly yields the gradient vector, facilitating an efficient training process. Ma et al. [20] put forward an approach where gradients are computed using the Hilbert-Schmidt Independence Criterion (HSIC). In this method, each network layer is optimized through block coordinate descent, achieved without gradient propagation. The objective is to maximize HSIC between a layer's activation and the target output, while minimizing HSIC between the layer's activation and the input. Kim et al. [18] propose the Deep Random Ferns (d-RFs) model, an efficient alternative to deep neural networks (DNNs). It adopts a layer-by-layer

optimization by randomized ensemble learning without backpropagation. This model streamlines classification task with fewer hyperparameters and lower complexity compared to DNNs.

4 Backpropagation-free LLM Training

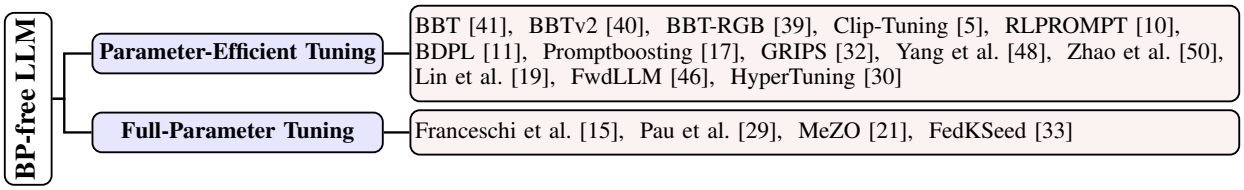

Figure 2: A taxonomy of BP-free LLM training.

Applying forward training methods to large language model (LLM) presents a myriad of challenges. One of the most significant issues is scalability. Current forward training techniques are adept at optimizing networks with lower-dimensional structures. However, their efficacy diminishes when applied to high-dimensional models, where they exhibit increased sensitivity and less robustness. This sensitivity to dimensionality complicates scaling efforts, as these methods struggle to maintain performance and accuracy in larger, more complex models. Furthermore, although forward training methods are more efficient than BP methods, multiple forward inferences still entail significant computational costs. Additionally, some methods require storing intermediate activations during forward inference, which leads to substantial memory consumption. Last but not least, it is crucial to consider whether BP-free training methods for LLM can be effectively compatible with low-power hardware. This adaptation is vital for enabling large-scale model training within resource-constrained environments. However, the feasibility of such implementations, given the inherent computational intensity and memory requirements of these large models, remains an open question. This challenge calls for further research in the field.

4.1 Parameter-Efficient Tuning

Given the sensitivity of forward training methods to high dimensionality, many studies focuse on their use for parameter-efficient fine-tuning (PEFT), updating only a subset of parameters in low-dimensional structures.

One of the most common approaches is prompt tuning. Sun et al. introduce a black-box tuning (BBT) [41] framework for derivative-free optimization of task-specific continuous prompts in a randomly generated subspace, demonstrating promising results in few-shot settings on the RoBERTa-LARGE backbone model. BBTv2 [40], an enhancement of BBT, introduces layer-wise prompts and utilizes a divide-and-conquer algorithm for their alternating optimization, enhancing versatility across various tasks and pre-trained models. To mitigate overfitting and local optima in few-shot settings encountered by previous methods, BBT-RGB [39] adopts a two-stage derivative-free optimization, Multi-Mixed verbalizers, and in-context learning integration, achieving more stable convergence. Clip-Tuning [5] further explores the diversity of rewards in derivative-free prompt optimization, using frozen subnetworks as multi-view critics for optimization instead of traditional evolutionary algorithms. RLPROMPT [10] diverges from the conventional focus on continuous prompts, employing reinforcement learning for the optimization of discrete prompts. It reveals the transferability of these prompts across various language models. Black-box Discrete Prompt Learning (BDPL) [11] highlights the security benefits of black-box prompt tuning for cloud infrastructure. It uses a variance-reduced policy gradient algorithm to efficiently optimize discrete prompts, suitable for commercial prediction APIs. Promptboosting [17] presents a novel approach for black-box prompt learning, focusing on optimizing the verbalizer paired with a single prompt through weak learners. This method achieves compatible performance in both few-shot and standard learning paradigms and offers a 10x speed improvement. Gradient-free Instructional Prompt Search (GRIPS) [32] doesn't generate prompts from scratch but instead employs an edit-based search approach to automatically rewrite prompts. Instead of optimizing traditional prompts, Yang et al. [48] emphasize leveraging the contextual learning capabilities of LLMs thus optimizing demonstrations through multiple forward passes. Furthermore, forward-based prompt tuning is gradually being applied in the federated settings [50][19].

In addition to prompt, forward training is also used to tune other additional network trainable modules. FwdLLM [46] combines the BP-free approach with trainable PEFT plugins (i.e., LoRA, Adapter, BitFit) in federated seetings. Edge devices perform forward propagation to obtain forward gradients of the trainable plugins and send them to the cloud for gradient updates. This represents the first implementation of fine-tuning a billion-parameter model (i.e., the 7B LLaMA) through a BP-free method, igniting the prospect of BP-free LLM training. In HyperTuning [30], rather than directly optimizing PEFT plugins (i.e., LoRA, soft prefix), the method updates a supplementary hypermodel. This

Model Name		BBT	BBTv2	BBT-RGB	Clip-Tuning	RLPROMPT	BDPL	Promptboosting	GRIPS	Yang et al.	Zhao et al.	Lin et al.	FwdLLM	HyperTuning	Franceschi et al.	Pau et al.	MeZO	FedKSeed
BERT	Distil (66M)												✓			✓		
	Base (110M)										✓		✓					
ALBERT	Large (340M)		✓	✓														
	base (12M)												✓					
RoBERTa	Base (125M)										✓							
	Large (355M)	✓	✓		✓		✓	✓					✓				✓	
GPT-2	distil (82M)					✓												
	Small (117M)					✓				✓								
	Medium (345M)					✓				✓								
	Large (774M)		✓			✓				✓								
	XL (1.5B)					✓			✓	✓								
	Small (125M)															✓		
GPT-3	Ada (350M)						✓											
	Babbage (1.3B)						✓											
	Curie (6.7B)						✓											
	Davinci (175B)						✓											
InstructGPT	Babbage (1.3B)								✓									
	Curie (6.7B)								✓									
E-GPT	Neo 125M									✓								
	Neo 1.3B									✓								
	Neo 2.7B									✓								
	J-6B									✓								
	NeoX-20B									✓								
LLaMA	3B																	✓
	7B												✓					
BART	Large (406M)		✓															
T5	base (220M)										✓							
	Large(770M)		✓											✓				
	XL (3B)													✓				
AlexaTM	19.75 B															✓		
CPM-2	(11B)		✓															
OPT	(125M)									✓								
	(350M)									✓								
	(1.3B)									✓								
	(2.7B)									✓								
	(6.7B)									✓								
	(13B)									✓							✓	
	30B																✓	
	66B																✓	
BLOOM	(560M)									✓								
	(1.1B)									✓								
	(1.7B)									✓								
	(3B)									✓								
	(7.1B)									✓								
CLIP ViT	vit-mae-base (86M)											✓						
Custom transformers (1∼8 layers)															✓			

Table 1: Model scope of different BP-free LLM training methods.

update is based on the output from forward propagation of a LLM model, which then enables the hypermodel to generate new parameters for these plugins.

4.2 Full-Parameter Tuning

As previously discussed, full-parameter fine-tuning through forward propagation presents challenges. However, certain studies have adopted specialized designs to implement effective methods even in resource-constrained environments.

Franceschi [15] pioneered the application of forward training within Transformer architecture models. This approach leverages the principles of target propagation to iteratively compute local objectives at each layer and effectuates local parameter updates via the mechanism of Local Representation Alignment (LRA). Pau et al. [29] extend the use of PEPITA and MEMPEPITA methods from simple models to Transformer-based models. This study tackles the issue of varying input and output dimensions especially in encoder-decoder architectures by employing attention mechanisms for error projection instead of fixed matrix. Additionally, the memory-efficient zerothorder optimizer (MeZO) [21] focuses on reducing memory consumption during zeroth-order optimization, which typically requires at least double the inference memory. It achieves this by resetting the seed to reconstruct the perturbation variables at each iteration, eliminating the need for continuous storage, thus reducing memory usage to near the inference memory. FedKSeed [33] is another gradient reconstruction method designed for federated scenarios. Instead of exchanged the gradients between clients and server directly, only seeds and scalar gradients are exchanged to reconstruct the real gradients. This method effectively reduces the data transmission costs in federated large model training.

5 Conclusions and future work

This survey provides a comprehensive overview of BP-free training for large foundation models. We have reviewed the background of BP-free training, including the motivation, challenges, and the taxonomy of BP-free training methods. We have also discussed the BP-free training methods, including the perturbated model, perturbated input, and no perturbabation methods. We have also discussed the applications of BP-free training, especially for large language model.

The research opportunity of BP-free training is extremely large, notably:

(1). **Scaling to larger models.** The BP-free training methods are still in the early stage, and the current methods are not yet mature enough to handle the training of mega models. The BP-free training methods need to be further optimized or specfically designed to handle the training of larger models.

(2). **Interagting current inference optimizations.** Though BP-free training is a promising direction, it is still not widely adopted in the industry. Many optimizations in the inference stage, such as quantization, pruning, and ealy-exit, are not yet integrated into BP-free training. From the perspective of hardware design, how to use the NPU to accelerate BP-free training is also a promising direction.

(3). **Recycling past inference results for BP-free training.** Enormous inference results are continuously generated in the deployment of large foundation models. If we could recycle these results for BP-free training, we could potentially significantly reduce the computational cost for task-specifici LLM fine-tuning.

(4). **BP-free Colloborative (Federated) Learning.** Collaborative learning, particularly in federated learning, encounters limitations due to high communication/computation costs and memory overhead on edge devices. BP-free training methods eliminate the need to store intermediate activations, making them memory-efficient. FwdLLM [46] is the first work to integrate BP-free training into federated learning, facilitating the training of billion-sized LLMs (like LLaMA) on commodity mobile devices.

References

[1] Florian Bacho and Dominique Chu. Low-variance forward gradients using direct feedback alignment and momentum, 2023.

[2] Atılım Güneş Baydin, Barak A Pearlmutter, Don Syme, Frank Wood, and Philip Torr. Gradients without backpropagation. *arXiv preprint arXiv:2202.08587*, 2022.

[3] Gabriel Belouze. Optimization without backpropagation, 2022.

[4] D. M. Bortz and C. T. Kelley. *The Simplex Gradient and Noisy Optimization Problems*, pages 77–90. Birkhäuser Boston, Boston, MA, 1998.

[5] Yekun Chai, Shuohuan Wang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. Clip-tuning: Towards derivative-free prompt learning with a mixture of rewards. *arXiv preprint arXiv:2210.12050*, 2022.

[6] Aochuan Chen, Yimeng Zhang, Jinghan Jia, James Diffenderfer, Jiancheng Liu, Konstantinos Parasyris, Yihua Zhang, Zheng Zhang, Bhavya Kailkhura, and Sijia Liu. Deepzero: Scaling up zeroth-order optimization for deep model training, 2024.

[7] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust Region Methods*. Society for Industrial and Applied Mathematics, 2000.

[8] Fabrizio De Vita, Rawan MA Nawaiseh, Dario Bruneo, Valeria Tomaselli, Marco Lattuada, and Mirko Falchetto. μ-ff: On-device forward-forward training algorithm for microcontrollers. In *2023 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 49–56. IEEE, 2023.

[9] G Dellaferrera and G Kreiman. Error-driven input modulation: Solving the credit assignment problem without a backward pass. arxiv 2022. *arXiv preprint arXiv:2201.11665*.

[10] Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric P Xing, and Zhiting Hu. Rlprompt: Optimizing discrete text prompts with reinforcement learning. *arXiv preprint arXiv:2205.12548*, 2022.

[11] Shizhe Diao, Zhichao Huang, Ruijia Xu, Xuechun Li, Yong Lin, Xiao Zhou, and Tong Zhang. Black-box prompt learning for pre-trained language models. *arXiv preprint arXiv:2201.08531*, 2022.

[12] Haozhe Feng, Tianyu Pang, Chao Du, Wei Chen, Shuicheng Yan, and Min Lin. Does federated learning really need backpropagation?, 2023.

[13] E. Fermi. Numerical solution of a minimum problem. 11 1952.

[14] Louis Fournier, Stéphane Rivaud, Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. Can forward gradient match backpropagation?, 2023.

[15] Dinko Franceschi. Backpropagation free transformers.

[16] Geoffrey Hinton. The forward-forward algorithm: Some preliminary investigations. *arXiv preprint arXiv:2212.13345*, 2022.

[17] Bairu Hou, Joe O'connor, Jacob Andreas, Shiyu Chang, and Yang Zhang. Promptboosting: Black-box text classification with ten forward passes. In *International Conference on Machine Learning*, pages 13309–13324. PMLR, 2023.

[18] Sangwon Kim and Byoung Chul Ko. Building deep random ferns without backpropagation. *IEEE Access*, 8:8533–8542, 2020.

[19] Zihao Lin, Yan Sun, Yifan Shi, Xueqian Wang, Lifu Huang, Li Shen, and Dacheng Tao. Efficient federated prompt tuning for black-box large pre-trained models. *arXiv preprint arXiv:2310.03123*, 2023.

[20] Wan-Duo Kurt Ma, JP Lewis, and W Bastiaan Kleijn. The hsic bottleneck: Deep learning without back-propagation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5085–5092, 2020.

[21] Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. *arXiv preprint arXiv:2305.17333*, 2023.

[22] Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D. Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes, 2024.

[23] Ali Momeni, Babak Rahmani, Matthieu Malléjac, Philipp Del Hougne, and Romain Fleury. Backpropagation-free training of deep physical neural networks. *Science*, 382(6676):1297–1303, 2023.

[24] Ilker Oguz, Junjie Ke, Qifei Wang, Feng Yang, Mustafa Yildirim, Niyazi Ulas Dinc, Jih-Liang Hsieh, Christophe Moser, and Demetri Psaltis. Forward-forward training of an optical neural network. *arXiv preprint arXiv:2305.19170*, 2023.

[25] Alexander Ororbia. Contrastive-signal-dependent plasticity: Forward-forward learning of spiking neural systems. *arXiv preprint arXiv:2303.18187*, 2023.

[26] Alexander Ororbia and Ankur Mali. The predictive forward-forward algorithm. *arXiv preprint arXiv:2301.01452*, 2023.

[27] Daniele Paliotta, Mathieu Alain, Bálint Máté, and François Fleuret. Graph neural networks go forward-forward. *arXiv preprint arXiv:2302.05282*, 2023.

[28] Danilo Pietro Pau and Fabrizio Maria Aymone. Suitability of forward-forward and pepita learning to mlcommons-tiny benchmarks. In *2023 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*, pages 1–6. IEEE, 2023.

[29] Danilo Pietro Pau and Fabrizio Maria Aymone. Forward learning of large language models by consumer devices. 13(2):402, 2024.

[30] Jason Phang, Yi Mao, Pengcheng He, and Weizhu Chen. Hypertuning: Toward adapting large language models without back-propagation. In *International Conference on Machine Learning*, pages 27854–27875. PMLR, 2023.

[31] Boris Polyak. *Introduction to Optimization*. 07 2020.

[32] Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. Grips: Gradient-free, edit-based instruction search for prompting large language models. *arXiv preprint arXiv:2203.07281*, 2022.

[33] Zhen Qin, Daoyuan Chen, Bingchen Qian, Bolin Ding, Yaliang Li, and Shuiguang Deng. Federated full-parameter tuning of billion-sized language models with communication cost under 18 kilobytes. *arXiv preprint arXiv:2312.06353*, 2023.

[34] Mengye Ren, Simon Kornblith, Renjie Liao, and Geoffrey Hinton. Scaling forward gradient with local losses, 2023.

[35] Riccardo Scodellaro, Ajinkya Kulkarni, Frauke Alves, and Matthias Schröter. Training convolutional neural networks with the forward-forward algorithm. *arXiv preprint arXiv:2312.14924*, 2023.

[36] David Silver, Anirudh Goyal, Ivo Danihelka, Matteo Hessel, and H. V. Hasselt. Learning by directional gradient descent. In *International Conference on Learning Representations*, 2022.

[37] Utkarsh Singhal, Brian Cheung, Kartik Chandra, Jonathan Ragan-Kelley, Joshua B. Tenenbaum, Tomaso A. Poggio, and Stella X. Yu. How to guess a gradient, 2023.

[38] James C. Spall. A stochastic approximation technique for generating maximum likelihood parameter estimates. In *1987 American Control Conference*, pages 1161–1167, 1987.

[39] Qiushi Sun, Chengcheng Han, Nuo Chen, Renyu Zhu, Jingyang Gong, Xiang Li, and Ming Gao. Make prompt-based black-box tuning colorful: Boosting model generalization from three orthogonal perspectives. *arXiv preprint arXiv:2305.08088*, 2023.

[40] Tianxiang Sun, Zhengfu He, Hong Qian, Yunhua Zhou, Xuan-Jing Huang, and Xipeng Qiu. Bbtv2: towards a gradient-free future with large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3916–3930, 2022.

[41] Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*, pages 20841–20855. PMLR, 2022.

[42] Virginia Torczon. On the convergence of the multidirectional search algorithm. *SIAM Journal on Optimization*, 1(1):123–145, 1991.

[43] Paul Vicol, Zico Kolter, and Kevin Swersky. Low-variance gradient estimation in unrolled computation graphs with es-single, 2023.

[44] Paul Vicol, Luke Metz, and Jascha Sohl-Dickstein. Unbiased gradient estimation in unrolled computation graphs with persistent evolution strategies, 2021.

[45] Bogdan M Wilamowski and Hao Yu. Neural network learning without backpropagation. *IEEE Transactions on Neural Networks*, 21(11):1793–1803, 2010.

[46] Mengwei Xu, Dongqi Cai, Yaozong Wu, Xiang Li, and Shangguang Wang. Fwdllm: Efficient fedllm using forward gradient, 2024.

[47] Mengwei Xu, Wangsong Yin, Dongqi Cai, Rongjie Yi, Daliang Xu, Qipeng Wang, Bingyang Wu, Yihao Zhao, Chen Yang, Shihe Wang, et al. A survey of resource-efficient llm and multimodal foundation models. *arXiv preprint arXiv:2401.08092*, 2024.

[48] Jiaxi Yang, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. Iterative forward tuning boosts in-context learning in language models. *arXiv preprint arXiv:2305.13016*, 2023.

[49] Liang Zhang, Kiran Koshy Thekumparampil, Sewoong Oh, and Niao He. Dpzero: Dimension-independent and differentially private zeroth-order optimization, 2023.

[50] Haodong Zhao, Wei Du, Fangqi Li, Peixuan Li, and Gongshen Liu. Fedprompt: Communication-efficient and privacy-preserving prompt tuning in federated learning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.