A Survey of Backpropagation-free Training For LLMS

Hanzi Mei[1], Dongqi Cai[1], Yaozong Wu[1], Shangguang Wang[1], and Mengwei Xu[1]

[1]Beijing University of Posts and Telecommunications (BUPT)

March 29, 2024

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 downstream tasks. However, training 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. We first outline three mainstream BP-free training methods. Subsequently, we introduce 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 [8]. These models have demonstrated unparalleled proficiency in a wide range of generic machine learning tasks [39], significantly advancing the capabilities of artificial intelligence. Traditionally, most LLMs, if not all, have been trained using a forward-backward paradigm [16, 18, 52, 49]. This paradigm predominantly relies on backpropagation (BP) to compute gradients, which are essential for updating the model weights [30]. Despite its effectiveness, BP-based methods encounter several limitations as the scale of models increases such as hugh resource costs, huge memory footprint, and incompatibility with device accelerators [69].

In light of these challenges, there has been a growing interest in exploring backpropagation-free (BP-free) training methods [36, 21, 11]. These alternative approaches primarily perform perturbed inference to derive the optimization directions, thereby circumventing the need for backpropagation computation to obtain exact gradients. While the majority of these methods were not initially designed for training LLMs, their inference-only nature leads to memory efficiency and compatibility with low-power devices, which is attractive for large-scale LLM training [69, 48, 61]. For example, the need of storing intermediate activations for backpropagation requires a 7.7 GB peak memory footprint for RoBERTa-large [32] training. In contrast, the inference of the same model only requires a 1.5 GB peak memory footprint, which is a 5.1x reduction.

Though BP-free training methods have shown promise in training LLMs, they are still in the early stages of development. One of the most significant challenges is the scalability of these methods to high-dimensional models, as they are more sensitive to dimensionality and less robust than BP-based methods [43]. Thus far, many optimizations have been proposed to address these challenges, such as tuning the low intrinsic dimension [1] of LLM [44, 34].

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. The flow of this survey is as follows: initially, we provide a brief overview of LLM and BP-free training methodsin §2; we then introduce three mainstream BP-free training methods and their variants in §3; subsequently, we introduce their optimizations for LLMs in §4; finally, we conclude the survey and propose several research directions in §5.

2 Background

2.1 Large Language Model

LLMs have marked a paradigm shift in the field of artificial intelligence [27], 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 [64] by Google. This architecture, distinguished by its self-attention mechanisms [37], 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.

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 [68], content generation [53], and image recognition [18].

The common workflow for training these LLMs involves a forward-backward process centered around 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 [59]. 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 [70]. 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 [50, 29]. 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.

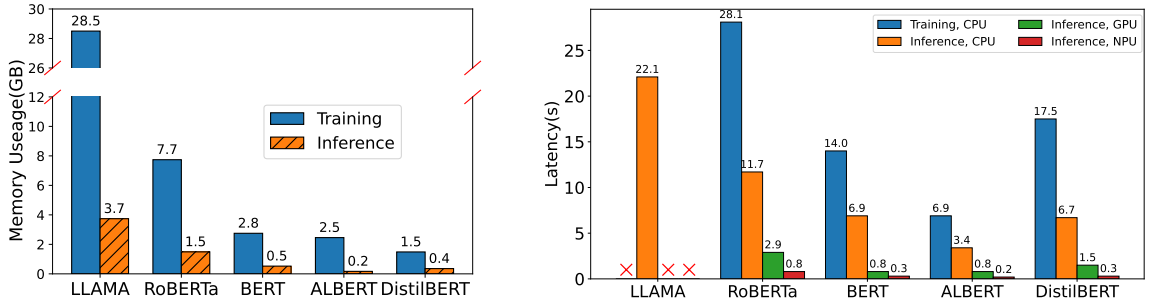
Incompatibility with Device Accelerators. Lastly, the training of LLMs is often incompatible with many common devices, like mobile phones, due to their incapacity to support such extensive computational and memory requirements [72]. This limitation significantly hinders the democratization and widespread 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 pursuit of deriving optimization directions for models without relying on backpropagation has garnered significant research interest. Early BP-free approaches, rooted in direct search, include coordinate search [20] and pattern search [63]. Additionally, model-based strategies like model-based descent [6] and trust region methods [12] have been explored. Evolutionary strategy iteratively refines a set of solutions to discover the optimal or near-optimal solution for a given problem, facilitating exploration of solution spaces to achieve desired optimization objectives. In recent years, perturbation-based forward propagation methods have garnered significant attention, notably the forward gradient, zeroth-order optimization and forward-forward techniques. forward gradient and zeroth-order optimization involve steering the function's descent along a randomly selected direction. This can be traced back to the earliest literature [46] and [58]. The inspiration for the forward-forward algorithm comes from Boltzmann machines [25] and noise contrastive estimation [23]. It processes the data input into the network and achieves model training through greedy multi-layer learning.

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 backpropagation. For instance, employing inference-only methods such as zeroth-order optimization results in up to $12.5\times$ memory reduction compared to BP-based methods, as illustrated in the Fig. 1a. Furthermore, the zeroth-order optimization method operates independently of modern deep learning software's automatic differentiation (AD) function, enhancing compatibility with inference accelerators deployed across diverse devices. These accelerators are specifically designed for inference, significantly speeding up the computation of inference-only optimization methods.



(a) Peak memory footprint

(b) Latency per batch

Figure 1: The comparison between training and inference. Batchsize: 8.

Fig. 1b illustrates the time taken for training and inference across different hardware devices, wherein edge-side GPUs and NPUs do not support model training.

3 Backpropagation-free Training Methods

We introduce three mainstream BP-free training methods and their variants 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. 2.

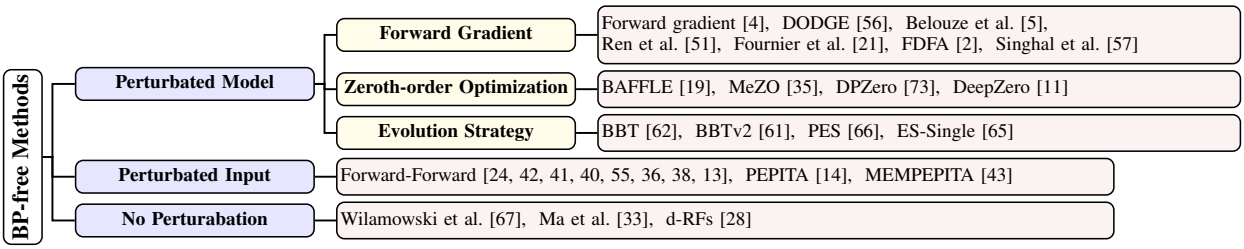


Figure 2: A taxonomy of BP-free training methods.

3.1 Perturbated Model

The theoretical foundation for BP-free training through perturbation of model parameters is categorized into three primary methods: forward mode AD, numerical differentiation, and evolution strategy. Among these, forward mode AD facilitates forward gradient computations, whereas numerical differentiation is chiefly employed for zeroth-order optimization.

Forward mode AD [3] applies symbolic differentiation to fundamental operators during forward propagation. By replacing parameter values and preserving intermediate derivatives, it calculates the gradient concerning parameters θ via the chain rule. This approach efficiently computes the Jacobian-vector product[1], facilitating forward gradient calculation. For a function $f : \mathbb{R}^n \to \mathbb{R}$ and parameters θ needing updates, the derivative of θ, denoted as $\dot{\theta}$, is initiated with a perturbation vector $v \sim \mathcal{N}(0, I)$. During a single forward propagation, the Jacobian-vector product shown in equation (1) represents the directional derivative of θ in the direction of v, obtained via AD. Baydin et al. [4] demonstrated that multiplying this derivative by the perturbation vector yields the forward gradient, shown in equation (2). The expected value of the forward gradient equates to the true gradient, making it directly applicable in gradient-based optimization methods. Forward-mode AD provides accurate gradient calculations in one forward pass but requires extra space for intermediate result storage.

$$J_f(\theta)\, v = \langle \nabla f(\theta), v \rangle = \sum_{i=1}^{n} \frac{\partial f}{\partial \theta_i} v_i \tag{1}$$

$$\hat{\nabla} f(\theta) = (J_f(\theta)\, v)\, v \tag{2}$$

[1]Can be implemented through the jvp operation in tensor frameworks such as JAX [7].

Numerical differentiation [9] employs a finite difference approximation, adding a perturbation $v \sim \mathcal{N}(0, I)$ to the parameter θ, to calculate an approximate gradient value from the objective function's resultant value. It is presented in two forms: forward (3) and central differences (4), aiming for ε to be nearly zero. This method, requiring two forward passes for gradient approximations, faces truncation errors and scalability issues for large parameter sets.

$$\hat{\nabla} f(\theta) = \frac{f(\theta + \varepsilon v) - f(\theta)}{\varepsilon} v \approx v v^T \nabla f(\theta) \tag{3}$$

$$\hat{\nabla} f(\theta) = \frac{f(\theta + \varepsilon v) - f(\theta - \varepsilon v)}{2\varepsilon} v \approx v v^T \nabla f(\theta) \tag{4}$$

Evolution strategy [54], a population-based heuristic search algorithm, excels in black-box optimization problem-solving. It starts with randomly sampling N Monte Carlo samples (particles) from a normal distribution $\mathcal{N}(0, I)$, with σ indicating perturbation intensity. Through parameter space perturbations, it estimates gradients, demonstrated in vanilla (5) and antithetic forms (6). The latter reduces estimate variance, improving accuracy. This strategy requires N or $2N$ forward passes for a single gradient estimation, depending on the employed form.

$$\hat{\nabla} f(\theta) = \frac{1}{\sigma N} \sum\nolimits_{i=1}^{N} (f(\theta + \sigma v_i)) v_i \tag{5}$$

$$\hat{\nabla} f(\theta) = \frac{1}{2\sigma N} \sum\nolimits_{i=1}^{N} (f(\theta + \sigma v_i) - f(\theta - \sigma v_i)) v_i \tag{6}$$

Gradient estimation can be achieved using all three previously mentioned methods, independent of backpropagation. This value can then be applied in gradient optimization methods, such as stochastic gradient descent (SGD) in equation(7), to update model parameters.

$$\theta = \theta - \eta \widehat{\nabla} f(\theta) \tag{7}$$

3.1.1 Forward Gradient

The forward gradient method, proposed by [4], has recently garnered widespread attention. They advocate employing forward mode AD 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. [56] 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 [5] argues that high-dimensional problems pose a significant challenge for forward gradients. In cases where optimization parameters are of exceedingly large dimensionality, forward gradients exhibit pronounced variance, resulting in substantial deviation from the true gradient.

To diminish the variance of forward gradients, Ren et al. [51] suggest 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. [21] propose 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 (DNNs), Bacho et al. [2] put 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. [57] explore 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

Zeroth-order optimization commonly employs numerical differentiation to approximate gradients. Feng et al. [19] introduce BAFFLE, a federated learning system devoid of backpropagation. BAFFLE utilizes zeroth-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. [35] present MeZO, a memory-efficient zeroth-order optimizer. MeZO employs fixed random seeds for in-place perturbations, resulting in memory

consumption comparable to inference during the optimization process. Combining zeroth-order optimization with differential privacy, Zhang et al. [73] propose the DPZero algorithm. Leveraging estimated gradients, DPZero exhibits convergence speed independent of dimensionality. Chen et al. [11] leverage zeroth-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

Evolution 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. [62] propose 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 [61] is an enhanced iteration of black box tuning (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. [66] present 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. [65] propose 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

Certain studies focus on perturbing the input data instead of altering the model. These approaches involve forward propagation not only with the original input data but also with its modified versions. The strategy for updating model parameters is based on the hidden layers' reactions to these two kinds of forward propagation.

The Forward-Forward (FF) Algorithm [24] proposes a training mechanism that utilizes two identical forward passes, differing only in the input data and their optimization objectives. The first pass processes positive (real) data aiming to maximize layer-wise 'goodness', while the second pass processes negative data with the goal of minimizing the same metric. The FF training approach has proven effective across various network architectures, including fully connected networks, Graph Neural Networks [42], Recurrent Neural Networks [41], Spiking Neural Networks [40], and Convolutional Neural Networks [55]. It is notably more suitable for low-power analog hardware than the BP algorithm. Adapted for resource-limited settings such as wave-based physical platforms [36, 38] and Micro-Controller Units (MCUs) [13], the FF algorithm enables neural network training beyond the traditional von Neumann architecture.

Moreover, the Present the Error to Perturb the Input To modulate Activity (PEPITA) method [14] views forward training as a credit assignment issue. Initially, it performs a standard forward propagation, then modulates the input data based on errors for a subsequent forward propagation. Neuronal updates are calculated from the differences between these two outcomes. Expanding on this, the MEMPEPITA algorithm [43] introduces a variant with three forward propagations, where activations and errors from the first pass are not stored. Instead, a third, standard forward propagation is conducted during the modulated pass. This method seeks memory efficiency, despite increasing computational demands.

3.3 No perturbation

Wilamowski et al. [67] present a BP-free training method, eliminating the need for perturbation. This approach extends the concept of signal gains, typically seen between neurons and outputs, to include inter-neuronal interactions. The direct computation of the signal gain matrix yields the gradient vector, enabling efficient training. Ma et al. [33] introduce a method that calculates gradients using the Hilbert-Schmidt Independence Criterion (HSIC). Here, each network layer is optimized through block coordinate descent, without gradient propagation, aiming to maximize HSIC between a layer's activation and the target output, and minimize HSIC between the layer's activation and the input. Kim et al. [28] propose the Deep Random Ferns (d-RFs) model, an efficient DNN alternative that employs layer-by-layer optimization via randomized ensemble learning without backpropagation, simplifying classification tasks with reduced hyperparameters and complexity.

4 Backpropagation-free LLM Training

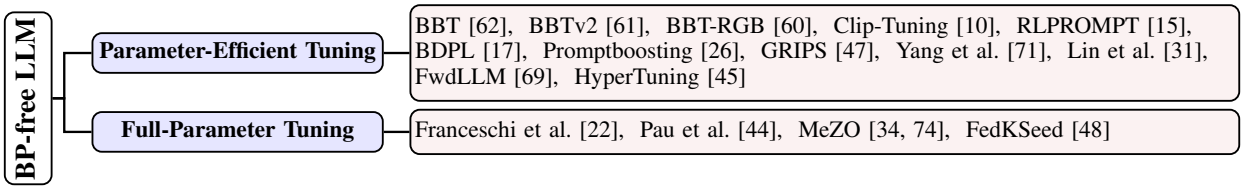


Figure 3: A taxonomy of BP-free LLM training.

Applying forward training methods to LLMs presents several challenges, with scalability being a primary concern. While current forward training techniques excel in optimizing networks with simpler, lower-dimensional structures, their effectiveness decreases when applied to high-dimensional models. In these cases, they show greater sensitivity and diminished robustness. This increased sensitivity complicates efforts to scale these methods, as they struggle to maintain performance and accuracy in larger and more complex models. Moreover, despite being more efficient than BP methods, multiple forward inferences incur significant computational costs. Additionally, some methods necessitate storing intermediate activations during forward inference, leading to high memory consumption. Furthermore, the compatibility of BP-free training methods with low-power hardware for LLMs is critical for facilitating large-scale model training in resource-limited settings. Yet, the practicality of such adaptations, considering the computational demands and memory needs of these extensive models, remains uncertain. This issue underscores the need for further research in the area.

4.1 Parameter-Efficient Tuning

Given the high-dimensionality sensitivity of forward training methods, numerous studies have focused on their application for Parameter-Efficient Fine-Tuning (PEFT), which updates only a subset of parameters in simpler structures.

Prompt tuning is a prevalent strategy. Sun et al. [62] introduced a BBT framework for optimizing task-specific continuous prompts in a randomly generated subspace, yielding promising few-shot results on the RoBERTa-large model. BBTv2 [61], an improvement over BBT, incorporates layer-wise prompts and a divide-and-conquer algorithm for alternating optimization, enhancing adaptability across different tasks and models. To overcome overfitting and local optima in few-shot settings, BBT-RGB [60] employs a two-stage derivative-free optimization, Multi-Mixed verbalizers, and in-context learning, achieving more stable convergence. Clip-Tuning [10] explores diverse rewards in derivative-free prompt optimization, utilizing frozen subnetworks as multi-view critics instead of conventional evolutionary algorithms. RLPROMPT [15] shifts from continuous to discrete prompts, using reinforcement learning for optimization and demonstrating their transferability across language models. Black-box Discrete Prompt Learning (BDPL) [17] emphasizes the security advantages of prompt tuning for cloud infrastructure, employing a variance-reduced policy gradient algorithm for efficient discrete prompt optimization, suitable for commercial APIs. Promptboosting [26] innovates in black-box prompt learning, optimizing the verbalizer with a single prompt via weak learners, offering comparable performance in both few-shot and standard paradigms with a significant speed increase. Gradient-free Instructional Prompt Search (GRIPS) [47] uses an edit-based search to automatically rewrite prompts, avoiding traditional prompt generation. Yang et al. [71] focus on leveraging LLMs' contextual learning through optimizing demonstrations via multiple forward passes. Additionally, forward-based prompt tuning is increasingly applied in federated settings [31].

Beyond prompts, forward training also tunes other network modules. FwdLLM [69] combines a BP-free approach with trainable PEFT plugins (e.g., LoRA, Adapter, BitFit) in federated settings, where edge devices forward propagate to obtain plugin gradients for cloud updates. This marks the first fine-tuning of a billion-parameter model (the 7B LLaMA) through a BP-free method, opening possibilities for BP-free LLM training. HyperTuning [45] updates a supplementary hypermodel based on LLM forward propagation output, enabling it to generate new parameters for PEFT plugins.

4.2 Full-Parameter Tuning

Despite challenges, certain studies have developed specialized designs for effective full-parameter fine-tuning through forward propagation, even in resource-limited environments.

Franceschi [22] pioneered forward training within Transformer models, using target propagation to compute layer-specific objectives and effectuate updates via Local Representation Alignment (LRA). Pau et al. [44] extended PEPITA

Model Name		BBT [62]	BBTv2 [61]	BBT-RGB [60]	Clip-Tuning [10]	RLPROMPT [15]	BDPL [17]	Promptboosting [26]	GRIPS [47]	Yang et al. [71]	Lin et al. [31]	FwdLLM [69]	HyperTuning [45]	Franceschi et al. [22]	Pau et al. [44]	MeZO [34]	FedKSeed [48]
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.

and MEMPEPITA methods to Transformer models, addressing varying input/output dimensions in encoder-decoder architectures with attention mechanisms for error projection. MeZO [34] focuses on reducing memory consumption in zeroth-order optimization by resetting the perturbation variable's seed at each iteration, thus eliminating continuous storage needs. Building upon the gradient estimation method proposed by MeZO, Zhang et al. [74] expanded gradient optimization beyond SGD to include sign-based SGD with 1-bit gradient quantization, momentum-based SGD, SGD with conservative gradient estimation, and the Adam optimizer. They then compared these methods to direct forward gradient training. FedKSeed [48], a gradient reconstruction method for federated settings, exchanges only seeds and scalar gradients to reconstruct real gradients, significantly reducing 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 models.

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). **Integrating 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, determining how to use NPUs 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 reduce the computational cost for task-specific LLM fine-tuning significantly.

(4). **BP-free Collaborative (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 [69] 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] Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020.

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

[3] Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of Marchine Learning Research*, 18:1–43, 2018.

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

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

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

[7] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, et al. Jax: Composable transformations of python+ numpy programs (v0. 2.5). *Software available from https://github. com/google/jax*, 2018.

[8] et al. Brown, Tom. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[9] Richard L Burden. *Numerical analysis*. Brooks/Cole Cengage Learning, 2011.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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*.

[15] 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.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[17] 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.

[18] et al. Dosovitskiy, Alexey. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

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

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

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

[22] Dinko Franceschi. Backpropagation free transformers.

[23] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

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

[25] Geoffrey E Hinton, Terrence J Sejnowski, et al. Learning and relearning in boltzmann machines. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1(282-317):2, 1986.

[26] 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.

[27] Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*, 2023.

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

[29] Taebum Kim, Hyoungjoo Kim, Gyeong-In Yu, and Byung-Gon Chun. Bpipe: Memory-balanced pipeline parallelism for training large language models. In *International Conference on Machine Learning*, pages 16639–16653. PMLR, 2023.

[30] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[31] 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.

[32] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62, 2021.

[38] 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.

[39] OpenAI. Gpt-4 technical report, 2023.

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

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

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

[43] 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.

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

[45] 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.

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

[47] 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.

[48] 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.

[49] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR, 2023.

[50] Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shuangyan Yang, Minjia Zhang, Dong Li, and Yuxiong He. {Zero-offload}: Democratizing {billion-scale} model training. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pages 551–564, 2021.

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

[52] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[53] Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021.

[54] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

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

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

[57] 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.

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

[59] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

[60] 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.

[61] 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.

[62] 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.

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

[64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

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

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

[68] Hongfei Xu, Josef van Genabith, Deyi Xiong, Qiuhui Liu, and Jingyi Zhang. Learning source phrase representations for neural machine translation. *arXiv preprint arXiv:2006.14405*, 2020.

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

[70] 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.

[71] 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.

[72] Jinliang Yuan, Chen Yang, Dongqi Cai, Shihe Wang, Xin Yuan, Zeling Zhang, Xiang Li, Dingge Zhang, Hanzi Mei, Xianqing Jia, et al. Rethinking mobile ai ecosystem in the llm era. *arXiv preprint arXiv:2308.14363*, 2023.

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

[74] Yihua Zhang, Pingzhi Li, Junyuan Hong, Jiaxiang Li, Yimeng Zhang, Wenqing Zheng, Pin-Yu Chen, Jason D Lee, Wotao Yin, Mingyi Hong, et al. Revisiting zeroth-order optimization for memory-efficient llm fine-tuning: A benchmark. *arXiv preprint arXiv:2402.11592*, 2024.