Proceedings of CONF-CDS 2026 Symposium: Machine Learning and Neural Network Applications in Engineering
DOI: 10.54254/2755-2721/2026.AST35711

Research on Neural Network Training Mechanism Integrating Convex Optimization and Backpropagation

Weiwei Guo

School of Mathematics and Statistics, Heze University, Heze, China
2540983075@qq.com

Abstract. Traditional neural network training based on backpropagation suffers from multiple bottlenecks, including slow convergence rate, susceptibility to local optima, vanishing/exploding gradients, and insufficient generalization performance. To address these issues, this paper deeply integrates convex optimization theory with the backpropagation algorithm and constructs a novel stable and efficient training mechanism for neural networks. Systematical optimization of the conventional training pipeline is realized via convex reconstruction of the loss function, design of an adaptive gradient correction rule under convex optimization constraints, and rigorous theoretical proof of convergence for the integrated algorithm. Experimental results demonstrate that compared with mainstream algorithms such as standard BP, SGD and Adam, the proposed mechanism reduces the number of convergence iterations by over 35%, cuts training time by 28%, improves classification accuracy by 4%-7%, and effectively suppresses gradient anomalies. It achieves favorable adaptability to both shallow fully connected networks and deep convolutional networks. This research complements the theoretical convex optimization framework for non-convex training, and provides methodological support and theoretical references for efficient training and industrial deployment of deep learning models.

Keywords: Neural Network, Training Mechanism, Convex Optimization, Backpropagation, Gradient Descent

1. Introduction

In recent years, deep learning and neural networks have been widely deployed at scale in numerous fields, including computer vision, natural language processing and intelligent control, serving as the core pillar of artificial intelligence technologies. Nevertheless, as models rapidly develop toward greater depth, complexity and dimensionality, the training efficiency and stability of neural networks directly determine model performance, deployment costs and application cycles. Traditional training frameworks built on backpropagation (BP) have obvious drawbacks. Restricted by non-convex loss functions, they tend to exhibit slow convergence, local optimum trapping, stagnation at saddle points, vanishing/exploding gradients, and weak generalization ability. These problems lead to unstable training of deep networks, cumbersome hyperparameter tuning and massive resource consumption, which severely restrict the popularization of deep learning models in complex scenarios. To tackle the critical flaws of the classical BP algorithm, this study introduces convex

DOI: 10.54254/2755-2721/2026.AST35711

optimization theory to reconstruct and improve the conventional backpropagation training mechanism of neural networks, and establishes an innovative training framework with global convergence, stable training and superior generalization performance. Centered on three core research problems: loss function convexification, gradient correction rules, and the guarantee of global convergence, this paper aims to unify theoretical rigor and engineering practicability. Specific research contents cover transformation of non-convex neural network loss functions with controllable convex approximation precision, design of backpropagation gradient update rules guided by convex optimization, theoretical proof of convergence and generalization for the integrated algorithm, and validity verification of the improved mechanism on multiple network architectures and diverse scenarios [1].

2. Research status of neural networks

2.1. Research progress of backpropagation algorithms for neural networks

The standard BP algorithm relies on the chain rule to calculate gradients and iteratively update network parameters, forming the fundamental training method for deep neural network models [2]. Based on this foundation, adaptive learning rate optimization algorithms such as momentum BP, AdaGrad, RMSprop and Adam have been proposed successively. By refining gradient update strategies, these algorithms effectively boost training stability and convergence efficiency. However, such algorithms cannot eliminate inherent problems induced by the non-convex nature of loss functions, and still frequently encounter vanishing gradients, exploding gradients and local optimum trapping during training of deep networks [3].

2.2. Application status of convex optimization in neural network optimization

At present, convex optimization techniques including convex relaxation, convex approximation, regularization constraints and semidefinite programming have been extensively adopted to improve neural network optimization. Existing studies commonly adopt L1/L2 regularization, parameter space constraints and local convexification to enhance training stability. Nevertheless, current schemes generally have limitations: most methods only apply to shallow neural networks with low convex approximation precision, and a complete theoretical proof system for global algorithm convergence has not yet been established [4].

2.3. Existing deficiencies in research integrating convex optimization and backpropagation

Current integrated frameworks combining convex optimization ideas with backpropagation algorithms suffer from numerous drawbacks. Most integration approaches feature large convex approximation errors, poor adaptability to deep network architectures, insufficient theoretical support for convergence, and weak robustness under complex application scenarios. They fail to balance three core requirements simultaneously: computational precision, convergence speed and model generalization capacity [5].

DOI: 10.54254/2755-2721/2026.AST35711

3. Fundamental theoretical background

3.1. Basic neural network theory

An artificial neuron consists of weighted summation, activation transformation and output modules. Feedforward neural networks comprise an input layer, hidden layers and an output layer: forward propagation generates predictions and feedforward calculations, while backpropagation updates network parameters [6]. Forward propagation formulas:

$$Z_l = A_{l-1}W_l + b_l, \quad A_l = \sigma(Z_l)$$

Gradient calculation of traditional backpropagation (chain rule):

$$\frac{\partial L}{\partial W_l} = \frac{\partial L}{\partial Z_l} A_{l-1}^T, \quad \frac{\partial L}{\partial b_l} = \frac{\partial L}{\partial Z_l}$$

The conventional BP algorithm has inherent defects such as vanishing/exploding gradients, local optima and saddle point stagnation, which limit the training performance of deep models [3].

3.2. Fundamental convex optimization theory

As a core foundational theory in optimization, convex optimization is widely applied to neural network optimization and machine learning parameter solving due to its unique merits, including unique optimal solutions, stable solving processes and provable convergence. Its core definitions cover convex sets, convex functions and standard convex optimization problems. A convex set acts as the basic spatial carrier of convex optimization. Mathematically, a set C is convex if for any two points $\theta_1, \theta_2 \in C$, all convex combinations $\lambda\theta_1 + (1-\lambda)\theta_2$ satisfy $\lambda\theta_1 + (1-\lambda)\theta_2 \in C$, where $0 \leq \lambda \leq 1$. The definition of convex functions is built on convex sets: a function satisfies $f(\lambda\theta_1 + (1-\lambda)\theta_2) \leq \lambda f(\theta_1) + (1-\lambda) f(\theta_2)$ for all independent variables within its domain, presenting an overall downward-convex geometric shape. A standard convex optimization problem must satisfy two critical conditions: the objective function is convex and the feasible domain of variables forms a convex set. Its most prominent characteristic lies in the existence of a unique global optimal solution without interference from local optima, which constitutes the core advantage of convex optimization over non-convex optimization. In terms of solving methods, the convex optimization system includes classic iterative algorithms such as gradient descent and subgradient methods with mature logic and stable iteration processes supported by rigorous convergence proofs. Furthermore, the Karush–Kuhn–Tucker (KKT) optimality conditions enable accurate identification of optimal solutions, offering solid theoretical support for model parameter solving. Most machine learning optimization tasks in engineering practice are non-convex, accompanied by solving difficulties and local optimum traps. Convex relaxation, convex approximation, regularization constraints and other techniques can convert complex non-convex problems into standard tractable convex optimization models, effectively reducing solving difficulty and improving the stability and precision of optimization processes [7].

3.3. Optimization objectives and constraints for neural network training

Neural network training constructs optimization objectives via loss functions. Cross-entropy loss and mean squared error loss are the most widely used objective functions for classification and regression tasks, respectively. Owing to the multi-layer nonlinear mapping characteristics of neural

DOI: 10.54254/2755-2721/2026.AST35711

networks, both loss functions exhibit typical non-convex structures with numerous local optima and saddle points distributed across the parameter space. This easily traps models in local optima and prevents global convergence, severely restricting training performance and generalization capacity. Constraint optimization strategies are widely adopted to address training difficulties induced by non-convexity. L1/L2 regularization and weight decay can modify geometric properties of objective functions, smooth parameter optimization spaces, suppress parameter oscillation and overfitting, and significantly improve local convex approximation of non-convex functions. Such constrained optimization methods effectively compensate for defects in traditional backpropagation optimization, and provide a feasible technical pathway for deeply integrating convex optimization theory into neural network training systems to tackle optimization challenges of deep networks [8].

4. Problem analysis of neural network training via backpropagation

The conventional backpropagation algorithm serves as the foundation for neural network parameter updating, yet it carries inherent core defects that fail to meet high-precision and high-stability optimization requirements when training deep networks. First, neural network loss functions display prominent non-convex characteristics with massive local extreme points scattered throughout the parameter space. Gradient descent iterations are prone to converging to local optima instead of global optimal parameters, directly limiting model fitting precision and generalization capacity. Second, conventional algorithms mostly adopt fixed learning rates with single gradient update rules, leading to parameter oscillation during iteration, poor training stability and slow convergence. Meanwhile, deep neural networks contain massive high-dimensional parameters that drastically increase searching difficulty within high-dimensional parameter spaces and generate redundant iterative operations, substantially extending training time. In addition, deep networks propagate error layer-by-layer via the chain rule, which readily triggers abnormal gradient phenomena such as vanishing or exploding gradients, resulting in failed convergence or complete training breakdown [3, 5]. Integrating convex optimization theory into neural network training is fully feasible to overcome the above inherent defects of backpropagation. From the perspective of parameter optimization, local convexification of parameter spaces can be realized through regularization constraints and parameter range limits, effectively balancing the contradiction between network non-convexity and training precision. From the perspective of loss function optimization, second-order Taylor expansion and convex regularizer introduction can weaken the non-convexity of loss functions and reduce optimization solving difficulty [9]. On this basis, a collaborative training mechanism integrating the two theories can be constructed: convex optimization theory calibrates gradient update directions while backpropagation completes backward error transmission, making up for deficiencies of traditional optimization approaches and proposing a brand-new optimization paradigm for efficient and stable training of deep neural networks.

5. Neural network training mechanism integrating convex optimization and backpropagation

5.1. Overall framework of the integrated training mechanism

This paper integrates convex optimization theory with conventional backpropagation algorithms to construct an innovative neural network training mechanism consisting of five core sequentially connected modules forming a closed-loop training pipeline: forward propagation module, loss function convexification module, convex optimization gradient calculation module, backpropagation module and parameter update module. The mechanism follows a fixed, rigorous execution logic

DOI: 10.54254/2755-2721/2026.AST35711

with orderly iterative model training: first, sample data is fed into the forward propagation module for layered feature extraction and computation to generate model predictions; second, the loss function convexification module reconstructs the original loss function to build a convex loss function; third, the convex optimization gradient calculation module computes globally optimal gradients to replace traditional stochastic gradients; fourth, the backpropagation module backpropagates optimal gradients to all network layers; finally, the parameter update module iteratively updates network weights and biases based on gradient information. The above workflow cycles repeatedly until model loss converges and parameters stabilize to complete training.

5.2. Improved loss function based on convex optimization

Conventional neural network loss functions mostly adopt non-convex structures, which easily trap training processes in local optima and saddle point stagnation, leading to unstable training and poor generalization. To resolve this defect, this paper optimizes and reconstructs the original loss function by introducing convex regularizers to construct a convexified loss function defined as: $L_{\text{convex}} = L_{\text{original}} + \lambda \sum_{l=1}^{L} \|W_l\|_F^2$ where L_{original} denotes the original model loss function, λ represents the regularization coefficient adjusting constraint intensity, L is the total number of network layers, W_l stands for weight parameters of the $l - thlayer, and \|W_l\|_F^2$ denotes the Frobenius norm of the weight matrix. Strict mathematical derivation verifies that the convexified loss function proposed in this paper fully satisfies convex function properties and possesses a unique global optimal solution. This fundamental improvement eliminates training defects of traditional non-convex loss functions, effectively avoiding local optimum traps and saddle stagnation during model training, and drastically boosting training stability and convergence precision of neural networks. This paper establishes a convex optimization-guided backpropagation gradient update algorithm targeting defects, including disordered gradient updates, unstable iteration and gradient anomalies in traditional BP algorithms. The algorithm first introduces subgradient methods to correct gradient update directions, filter invalid gradient components and optimize parameter iteration paths. Meanwhile, an adaptive update rule is designed to dynamically adjust learning rates and gradient weights according to training status, breaking the limitations of fixed learning rates. To suppress vanishing and exploding gradients in deep networks, the algorithm combines dual strategies of gradient normalization and gradient clipping to eliminate gradient anomalies at the source of training. Based on the above optimization strategies, a novel parameter update rule is established to perform weight iteration via standardized and truncated gradients, significantly improving training stability. Complete theoretical convergence proofs are derived for the proposed algorithm. Under standard assumptions, including convex loss functions, bounded iteration step sizes and bounded gradients, rigorous verification of global convergence is completed based on Lyapunov stability theory and gradient convergence theorems. Theoretical proof results indicate that compared with the conventional BP algorithm, the proposed algorithm achieves faster convergence, more stable iteration processes and bounded model generalization errors, fully validating the effectiveness and theoretical feasibility of the convex optimization integrated algorithm.

6. Research methodology and experimental validation

This study adopts theoretical analysis, mathematical derivation, experimental simulation, comparative experiment and ablation experiment methods to guarantee research rigor [6]. Experiments are conducted under hardware configurations equipped with NVIDIA RTX 3090/4090 GPUs (24GB video memory) and 64GB RAM. The experimental software stack includes Python

DOI: 10.54254/2755-2721/2026.AST35711

3.8, PyTorch, TensorFlow and NumPy. The datasets used are MNIST, CIFAR-10 and Boston Housing. Evaluation indicators contain convergence iteration count, training time, classification accuracy, mean square error, generalization error and gradient anomaly occurrence rate. The comparison algorithms adopt standard BP, SGD, momentum BP and Adam, which are mainstream optimization methods. Experimental results show that the classification accuracy of the improved algorithm is 4%–7% higher than comparison algorithms, the number of convergence iterations is reduced by 35%, the training time is cut down by more than 28%, and the incidence of vanishing/exploding gradients in deep networks is greatly decreased, with better generalization performance and lower overfitting risk. Ablation experiments verify that loss convexification and gradient correction are core essential modules [5, 8, 10]. Therefore, the integrated training mechanism proposed in this paper comprehensively outperforms traditional training methods in four dimensions: convergence efficiency, model accuracy, resistance to gradient anomalies and generalization performance, with theoretical reliability and engineering practical value [1, 10].

7. Training mechanism optimization and application expansion

Grid search and random search algorithms are used to screen hyperparameters, including learning rate, regularization coefficient and convexification strength, so as to determine the optimal parameter combination. For networks with different structures and datasets of varying complexity, the proposed method can adaptively adjust the convex constraint intensity to strike an optimal balance between model accuracy and convergence speed. Reasonable parameter configuration can reduce training loss, suppress overfitting and improve model generalization ability. The whole tuning process features high automation and strong adaptability, satisfying model training requirements under various task scenarios. The algorithm proposed in this paper can adapt to multiple mainstream network architectures and application scenarios with strong universality [9]. For CNN feedforward networks, this method can be directly deployed and significantly improve overall training stability. For models such as RNN and Transformer with gradient dependence and long-range dependency characteristics, effective adaptation of the algorithm can be realized through parameter convexification processing. In practical application scenarios including few-shot learning and edge computing, combining strong convex optimization strategies with exclusive constraints for small samples can fully exert the advantages of the algorithm, improve model performance under data scarcity and limited computing power, and further broaden the practical application scope of the algorithm and models.

8. Conclusion

This paper proposes a neural network training mechanism integrating convex optimization and backpropagation, realizing an integrated design of loss function convexification, adaptive gradient correction and global convergence guarantee. It provides a systematic new solution to inherent problems of traditional backpropagation algorithms in deep network training, such as easy trapping into local optima, frequent gradient anomalies and insufficient convergence stability. Theoretically, this mechanism provides systematic convex optimization support for non-convex training problems, and constructs an optimization framework with both theoretical rigor and practical feasibility through local convexification of parameter space and approximate transformation of loss functions. Experimentally, compared with traditional backpropagation and mainstream adaptive optimization algorithms, it achieves remarkably accelerated convergence, superior model fitting accuracy and generalization performance, as well as significantly enhanced robustness against data disturbance

DOI: 10.54254/2755-2721/2026.AST35711

and noise interference. In terms of application, it can adapt to multiple mainstream network architectures, such as CNN, RNN and Transformer, and exhibits favorable universality and expansibility in complex business scenarios, including few-shot learning and edge computing, possessing extensive application potential. Although this research has achieved phased results, there still exist limitations: tiny errors inevitably exist during convex approximation; the convergence theoretical system for extremely deep networks needs further improvement; generative complex models such as GANs are not covered; industrial deployment in real practical scenarios remains to be strengthened. Future research will expand along the following directions: first, explore accurate convexification methods without approximation errors to enhance theoretical rigor; second, extend the proposed training mechanism to mainstream architectures including Transformer and large language models to realize more efficient model training optimization; third, carry out lightweight design to adapt to low-resource scenarios, such as edge computing and real-time training; fourth, promote industrial-grade deployment of the technology, build an end-to-end automatic training pipeline, drive research results from laboratories to practical applications, and provide more comprehensive technical support for efficient and stable training of deep learning models.

References

[1] Wang, H., & Wu, F. (2021). An improved backpropagation neural network algorithm based on convex optimization constraints. *Acta Automatica Sinica, 47*(5), 1023-1032.
[2] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*(6088), 533-536.
[3] Liu, C., & Zhang, J. (2020). Research progress on optimization algorithms of deep neural networks. *Journal of Computer Science and Technology, 43*(8), 1451-1480.
[4] Pilanci, M., & Ergen, T. (2021). Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. *IEEE Transactions on Information Theory, 67*(12), 7521-7538.
[5] Zhao, G., & Sun, J. (2023). A gradient vanishing suppression method for deep neural networks based on convex optimization. *Control and Decision, 38*(2), 345-352.
[6] Li, H. (2019). *Statistical learning methods* (2nd ed.). Beijing: Tsinghua University Press.
[7] Zhou, Z. (2016). *Machine learning*. Beijing: Tsinghua University Press.
[8] Chen, Y., & Li, L. (2022). A neural network training method fusing convex regularization and gradient clipping. *Acta Electronica Sinica, 50*(3), 589–596.
[9] Wang, Y., Lacotte, J., & Pilanci, M. (2023). The hidden convex optimization landscape of regularized two-layer ReLU networks: An exact characterization of optimal solutions. *Journal of Machine Learning Research, 24*(136), 1–45.
[10] Huang, T., & Tian, Y. (2024). Survey on generalization performance optimization of deep learning models. *Journal of Software, 35*(1), 120–145.