Neurocomputing 487 (2022) 130–143

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Accelerated Gradient-free Neural Network Training by Multi-convex Alternating Optimization

Junxiang Wang [a], Hongyi Li [b], Liang Zhao [a,*]

[a] Emory University, 201 Dowman Dr, Atlanta, GA 30322, USA

[b] The State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, Shannxi 710071, China

ARTICLE INFO

Article history:
Received 26 September 2021
Revised 20 December 2021
Accepted 11 February 2022
Available online 25 February 2022

Keywords:
Deep learning
Alternating minimization
Nesterov acceleration
Linear convergence

ABSTRACT

In recent years, even though Stochastic Gradient Descent (SGD) and its variants are well-known for training neural networks, it suffers from limitations such as the lack of theoretical guarantees, vanishing gradients, and excessive sensitivity to input. To overcome these drawbacks, alternating minimization methods have attracted fast-increasing attention recently. As an emerging and open domain, however, several new challenges need to be addressed, including 1) Convergence properties are sensitive to penalty parameters, and 2) Slow theoretical convergence rate. We, therefore, propose a novel monotonous Deep Learning Alternating Minimization (mDLAM) algorithm to deal with these two challenges. Our innovative inequality-constrained formulation infinitely approximates the original problem with non-convex equality constraints, enabling our convergence proof of the proposed mDLAM algorithm regardless of the choice of hyperparameters. Our mDLAM algorithm is shown to achieve a fast linear convergence by the Nesterov acceleration technique. Extensive experiments on multiple benchmark datasets demonstrate the convergence, effectiveness, and efficiency of the proposed mDLAM algorithm.

1. Introduction

Stochastic Gradient Descent (SGD) and its variants have become popular optimization methods for training deep neural networks. Many variants of SGD methods have been presented, including SGD momentum [25], AdaGrad [8], RMSProp [28], Adam [12] and AMSGrad [19].While many researchers have provided solid theoretical guarantees on the convergence of SGD [12,19,25], the assumptions of their proofs cannot be applied to problems involving deep neural networks, which are highly nonsmooth and nonconvex. Aside from the lack of theoretical guarantees, several additional drawbacks restrict the applications of SGD. It suffers from the gradient vanishing problem, meaning that the error signal diminishes as the gradient is backpropagated, which prevents the neural networks from utilizing further training [27], and the gradient of the activation function is highly sensitive to the input (i.e. poor conditioning), so a small change in the input can lead to a dramatic change in the gradient.

To tackle these intrinsic drawbacks of gradient descent optimization methods, alternating minimization methods have started to attract attention as a potential way to solve deep learning problems. A neural network problem is reformulated as a nested function associated with multiple linear and nonlinear transformations across multi-layers. This nested structure is then decomposed into a series of linear and nonlinear equality constraints by introducing auxiliary variables and penalty hyperparameters. The linear and nonlinear equality constraints generate multiple subproblems, which can be minimized alternately. Many recent alternating minimization methods have focused on applying the Alternating Direction Method of Multipliers (ADMM) [27,31], Block Coordinate Descent (BCD) [37] and Method of Auxiliary Coordinates (MAC) [4] to replace a nested neural network with a constrained problem without nesting, with empirical evaluations demonstrating good scalability in terms of the number of layers and high accuracy on the test sets. These methods also avoid gradient vanishing problems and allow for non-differentiable activation functions such as binarized neural networks [7], as well as allowing for complex non-smooth regularization and the constraints that are increasingly important for deep neural architectures that are required to satisfy practical requirements such as interpretability, energy-efficiency, and cost awareness [4]. The ADMM, as a representative of alternating minimization methods, has been explored extensively for different neural network architectures. It was first used to solve the Multi-Layer Perceptron (MLP) problem with convergence

* Corresponding author.
E-mail addresses: jwan936@emory.edu (J. Wang), lihongyi@stu.xidian.edu.cn (H. Li), lzhao41@emory.edu (L. Zhao).

https://doi.org/10.1016/j.neucom.2022.02.039
0925-2312/

guarantees [27,31], and then was extended to other architectures such as Recurrent Neural Network (RNN) [26]. Recently, it was utilized to achieve parallel neural network training [29,30].

However, as an emerging domain, alternating minimization for deep model optimization suffers from several unsolved challenges including **1. Convergence properties are sensitive to penalty parameters.** One recent work by Wang et al. firstly proved the convergence guarantee of ADMM in the MLP problem [31]. However, such convergence guarantee is dependent on the choice of penalty hyperparameters: the convergence cannot be guaranteed anymore when penalty hyperparameters are small; **2. Slow convergence rate.** To the best of our knowledge, almost all existing alternating minimization methods can only achieve a sublinear convergence rate. For example, The convergence rate of the ADMM and the BCD is proven to be $O(1/k)$, where k is the number of iterations [31,37]. Therefore, there is still a lack of a theoretical framework that can achieve a faster convergence rate.

To simultaneously address these technical problems, we propose a new formulation of the neural network problem, along with a novel monotonous Deep Learning Alternating Minimization (mDLAM) algorithm. Specifically, we, for the first time, transform the original neural network optimization problem into an inequality-constrained problem that can infinitely approximate the original one. Applying this innovation to an inequality-constraint-based transformation not only ensures the convexity and hence easily ensures global minima of all subproblems, but also prevents the output of a nonlinear function from changing much and reduces sensitivity to the input. Moreover, our proposed mDLAM algorithm can achieve a linear convergence rate theoretically, and the choice of hyperparameters does not affect the convergence of our mDLAM algorithm theoretically. Extensive experiments on four benchmark datasets show the convergence, effectiveness, and efficiency of the proposed mDLAM algorithm. Our contributions in this paper include:

- We propose a novel formulation for neural network optimization. The deeply nested activation functions are disentangled into separate functions innovatively coordinated by inherently convex inequality constraints.
- We present an efficient optimization algorithm. A quadratic approximation technique is utilized to avoid matrix inversion. Every subproblem has a closed-form solution. The Nesterov acceleration technique is applied to further boost convergence.
- We investigate the convergence of the proposed mDLAM algorithm under mild conditions. The new mDLAM algorithm is guaranteed to converge to a stationary point no matter what hyperparameters we choose. Furthermore, the proposed mDLAM algorithm is shown to achieve a linear convergence rate, which is faster than existing methods.
- Extensive experiments have been conducted to demonstrate the effectiveness of the proposed mDLAM algorithm. We test our proposed mDLAM algorithm on four benchmark datasets. Experimental results illustrate that our proposed mDLAM algorithm is linearly convergent on four datasets, and outperforms consistently state-of-the-art optimizers. Sensitivity analysis on the running time shows that it increases linearly with the increase of hidden units and hyperparameters.

The rest of this paper is organized as follows: In Section 2, we summarize recent related research work to this paper. In Section 3, we formulate the MLP training problem and present the proposed mDLAM algorithm to train the MLP model. Section 4 details the convergence properties of the proposed mDLAM algorithm. Extensive experiments on benchmark datasets are shown in Section 5, and Section 6 concludes this work.

2. Related work

All existing works on deep learning optimization methods fall into two major categories: SGD methods and alternating minimization methods, which are shown as follows:

SGD methods: The renaissance of SGD can be traced back to 1951 when Robbins and Monro published the first paper [20]. The famous back-propagation algorithm was introduced by Rumelhart et al. [22]. Many variants of SGD methods have since been presented, including the use of Polyak momentum, which accelerates the convergence of iterative methods [17], and research by Sutskever et al., who highlighted the importance of Nesterov momentum and initialization [25]. During the last decade, many well-known SGD methods which are incorporated with adaptive learning rates have been proposed by the deep learning community, which include but are not limited to AdaGrad [8], RMSProp [28], Adam [12], AMSGrad [19], Adabelief [41] and Adabound [15].

Applications of alternating minimization methods for deep learning: Many recent works have applied alternating minimization algorithms to specific deep learning applications. For example, Taylor et al. and Wang et al. presented the ADMM to solve an MLP training problem via transforming it into an equality-constrained problem, where many subproblems split by ADMM can be solved efficiently [27,31], Wang et al. proposed a parallel ADMM algorithm to train deep MLP models [29], and a similar algorithm was extended to Graph Augmented-MLP (GA-MLP) models with the introduction of the quantization technique [30]. Zhang et al. handled Very Deep Supervised Hashing (VDSH) problems by utilizing an ADMM algorithm to overcome issues related to vanishing gradients and poor computational efficiency [40]. Zhang and Bastiaan trained a deep neural network by utilizing ADMM with a graph [38] and Askari et al. introduced a new framework for MLP models and optimize the objective using BCD methods [1]. Li et al. proposed an ADMM algorithm to achieve distributed learning of Graph Convolutional Network (GCN) via community detection [14]. Qiao et al. proposed an inertial proximal alternating minimization to train MLP models [18].

Convergence of alternating minimization methods for deep learning: Aside from applications, the other branch of works mathematically proves the convergence of the proposed alternating minimization approaches. For instance, Carreira and Wang proposed a method involving the use of auxiliary coordinates to replace a nested neural network with a constrained problem without nesting [4]. Lau et al. proposed a BCD optimization framework and proved the convergence via the Kurdyka-Lojasiewicz (KL) property [13], while Choromanska et al. proposed a BCD algorithm for training deep MLP models based on the concept of co-activation memory [6], and a BCD algorithm with R-linear convergence was proposed by Zhang and Brand to train Tikhonov regularized deep neural networks [39]. Jagatap and Hegde introduced a new family of alternating minimization methods and prove their convergence to a global minimum [11]. Yu et al. proved the convergence of the proposed ADMM for RNN models [26]. However, to the best of our knowledge, there is a lack of a flexible framework which allows for different activation functions and guarantees a linear convergence rate.

3. Model and algorithms

3.1. Inequality approximation for deep learning

Important notations used in this paper are shown in Table 1. A typical MLP model consists of L layers, each of which are defined by a linear mapping and a nonlinear activation function. A linear map-

Table 1
Notations used in this paper.

Notations	Descriptions
L	Number of layers.
W_l	The weight vector in the l-th layer.
z_l	The output of the linear mapping in the l-th layer.
$h_l(z_l)$	The nonlinear activation function in the l-th layer.
a_l	The output of the l-th layer.
x	The input matrix of the neural network.
y	The predefined label vector.
$R(z_L; y)$	The loss function in the L-th layer.
$\Omega_l(W_l)$	The regularization term in the l-th layer.
ε	The tolerance of the nonlinear mapping.

ping is composed of a weight vector $W_l \in \mathbb{R}^{n_l \times n_{l-1}}$, where n_l is the number of neurons on the l-th layer; a nonlinear mapping is defined by a continuous activation function $h_l(\bullet)$. Given an input $a_{l-1} \in \mathbb{R}^{n_{l-1}}$ from the $(l-1)$-th layer, the l-th layer outputs $a_l = h_l(W_l a_{l-1})$. By introducing an auxiliary variable z_l as the output of the linear mapping, the neural network problem is formulated mathematically as follows:

Problem 1.

$$\min_{a_l, W_l, z_l} R(z_L; y) + \sum_{l=1}^{L} \Omega_l(W_l),$$
$$s.t.\ z_l = W_l a_{l-1}\ (l = 1, \cdots, L),\ a_l = h_l(z_l)\ (l = 1, \cdots, L-1),$$

where $a_0 = x \in \mathbb{R}^d$ is the input of the neural network, d is the number of feature dimensions, and y is a predefined label vector. $R(z_L; y) \geqslant 0$ is a continuous loss function for the L-th layer, which is convex and proper, and $\Omega_l(W_l) \geqslant 0$ is a regularization term on the l-th layer, which is also continuous, convex and proper.

The equality constraint $a_l = h_l(z_l)$ is the most challenging one to handle here because common activation functions such as sigmoid are nonlinear. This makes them nonconvex constraints and hence it is difficult to obtain the optimal solution when solving the z_l-subproblem [27]. Moreover, there is no guarantee for alternating minimization methods to solve the nonlinear equality constrained Problem 1 [32]. To deal with these two challenges, the following assumption is required for problem transformation:

Assumption 1. $h_l(z_l)\ (l = 1, \ldots, n)$ are quasilinear.

The quasilinearity is defined in the appendix. Assumption 1 is so mild that most of the widely used nonlinear activation functions satisfy it, including tanh [35], smooth sigmoid [10], and the Rectified Linear Unit (ReLU) [16]. Then we innovatively transform the original nonconvex constraints into inequality constraints, which can be an infinite approximation of Problem 1. To do this, we introduce a tolerance $\varepsilon > 0$ and reformulate Problem 1 to the following:

$$\min_{W_l, z_l, a_l} R(z_L; y) + \sum_{l=1}^{L} \Omega_l(W_l)$$
$$s.t.\ z_l = W_l a_{l-1} (l = 1, \cdots, L),\ h_l(z_l) - \varepsilon \leqslant a_l \leqslant h_l(z_l) + \varepsilon (l = 1, \cdots, L-1).$$

For the linear constraint $z_l = W_l a_{l-1}$, this can be transformed into a penalty term in the objective function to minimize the difference between z_l and $W_l a_{l-1}$. The formulation is shown as follows:

Problem 2.

$$\min_{W_l, z_l, a_l} F(\mathbf{W}, \mathbf{z}, \mathbf{a}) = R(z_L; y) + \sum_{l=1}^{L} \Omega_l(W_l) + \sum_{l=1}^{L} \phi(a_{l-1}, W_l, z_l)$$
$$s.t.\ h_l(z_l) - \varepsilon \leqslant a_l \leqslant h_l(z_l) + \varepsilon (l = 1, \cdots, L-1).$$

The penalty term is defined as $\phi(a_{l-1}, W_l, z_l) = \frac{\rho}{2}\|z_l - W_l a_{l-1}\|_2^2$, where $\rho > 0$ a penalty parameter. $\mathbf{W} = \{W_l\}_{l=1}^{L}, \mathbf{z} = \{z_l\}_{l=1}^{L}$, $\mathbf{a} = \{a_l\}_{l=1}^{L-1}$. As $\rho \to \infty$ and $\varepsilon \to 0$, Problem 2 approaches Problem 1.

The introduction of ε is to project the nonconvex constraints to ε-balls, thus transforming the nonconvex Problem 1 into Problem 2. Even though Problem 2 is still nonconvex because $h_l(z_l)$ can be nonconvex (e.g. tanh and smooth sigmoid), it is convex with regard to one variable when others are fixed (i.e. multi-convex), which is much easier to solve by alternating minimization [34]. For example, Problem 2 is convex with regard to $\mathbf{z}$ when $\mathbf{W}$, and $\mathbf{a}$ are fixed.

Algorithm 1: the proposed mDLAM algorithm

Input: $y, a_0 = x$. **Output**: $a_l, W_l, z_l (l = 1, \cdots, L)$.
Initialize $\rho, k = 0.\ s^0 = 0$. **repeat**
$s^{k+1} \leftarrow \frac{1+\sqrt{1+4(s^k)^2}}{2}$
$\omega^k \leftarrow \frac{s^k - 1}{s^{k+1}}$
for $l = 1$ to L **do**
$\overline{W}_l^{k+1} \leftarrow W_l^k + \left(W_l^k - W_l^{k-1}\right)\omega^k$ and update W_l^{k+1} in Eq. (3).
if $F\left(\mathbf{W}_{\leqslant l}^{k+1}, \mathbf{z}_{\leqslant l-1}^{k+1}, \mathbf{a}_{\leqslant l-1}^{k+1}\right) \geqslant F\left(\mathbf{W}_{\leqslant l-1}^{k+1}, \mathbf{z}_{\leqslant l-1}^{k+1}, \mathbf{a}_{\leqslant l-1}^{k+1}\right)$ #W_l^{k+1} increases the objective F **then**
$\overline{W}_l^{k+1} \leftarrow W_l^k$ and update W_l^{k+1} in Eq. (3).
end if
$\overline{z}_l^{k+1} \leftarrow z_l^k + (z_l^k - z_l^{k-1})\omega^k$
if $l = L$ **then**
Update z_L^{k+1} in Eq. (5).
if $F\left(\mathbf{W}_{\leqslant L}^{k+1}, \mathbf{z}_{\leqslant L}^{k+1}, \mathbf{a}_{\leqslant L-1}^{k+1}\right) \geqslant F\left(\mathbf{W}_{\leqslant L}^{k+1}, \mathbf{z}_{\leqslant L-1}^{k+1}, \mathbf{a}_{\leqslant L-1}^{k+1}\right)$ #z_L^{k+1} increases the objective F **then**
$\overline{z}_L^{k+1} \leftarrow z_L^k$ and update z_L^{k+1} in Eq. (5).
end if
else
Update z_l^{k+1} in Eq. (4).
if $F\left(\mathbf{W}_{\leqslant l}^{k+1}, \mathbf{z}_{\leqslant l}^{k+1}, \mathbf{a}_{\leqslant l-1}^{k+1}\right) \geqslant F\left(\mathbf{W}_{\leqslant l}^{k+1}, \mathbf{z}_{\leqslant l-1}^{k+1}, \mathbf{a}_{\leqslant l-1}^{k+1}\right)$ #z_l^{k+1} increases the objective F **then**
$\overline{z}_l^{k+1} \leftarrow z_l^k$ and update z_l^{k+1} in Eq. (4).
end if
$\overline{a}_l^{k+1} \leftarrow a_l^k + (a_l^k - a_l^{k-1})\omega^k$ and update a_l^{k+1} in Eq. (6).
if $F\left(\mathbf{W}_{\leqslant l}^{k+1}, \mathbf{z}_{\leqslant l}^{k+1}, \mathbf{a}_{\leqslant l}^{k+1}\right) \geqslant F\left(\mathbf{W}_{\leqslant l}^{k+1}, \mathbf{z}_{\leqslant l}^{k+1}, \mathbf{a}_{\leqslant l-1}^{k+1}\right)$ #a_l^{k+1} increases the objective F **then**
$\overline{a}_l^{k+1} \leftarrow a_l^k$ and update a_l^{k+1} in Eq. (6).
end if
end if
end for
$k \leftarrow k + 1$.
until convergence.
Output a_l, W_l, z_l.

3.2. Alternating optimization

We present the mDLAM algorithm to solve Problem 2 in this section. A potential challenge to solve Problem 2 is a slow theoretical convergence rate. For example, the convergence rate of the dlADMM algorithm to solve Problem 2 is sublinear $o(1/k)$, where k is the number of iterations [31]. In order to address this challenge, we apply the famous Nesterov acceleration technique to boost the convergence of our proposed mDLAM algorithm, and we prove its linear convergence theoretically in the next section.

Algorithm 1 shows our proposed mDLAM algorithm. To simplify the notation, $\mathbf{W}^{k+1}_{\leqslant l} = \left\{\left\{W^{k+1}_i\right\}^l_{i=1}, \left\{W^k_i\right\}^L_{i=l+1}\right\}, \mathbf{z}^{k+1}_{\leqslant l} = \left\{\{z^{k+1}_i\}^l_{i=1}, \{z^k_i\}^L_{i=l+1}\right\}$ and $\mathbf{a}^{k+1}_{\leqslant l} = \left\{\{a^{k+1}_i\}^l_{i=1}, \{a^k_i\}^{L-1}_{i=l+1}\right\}$. In Algorithm 1, Lines 6, 10, and 21 apply the Nestrov acceleration technique and update W_l, z_l and a_l, respectively. the proposed mDLAM algorithm guarantees the decrease of objective F: for example, if the updated W^{k+1}_l in Line 7 of Algorithm 1 increases the value of F, i.e. $F\left(\mathbf{W}^{k+1}_{\leqslant l}, \mathbf{z}^{k+1}_{\leqslant l-1}, \mathbf{a}^{k+1}_{\leqslant l-1}\right) \geqslant F\left(\mathbf{W}^{k+1}_{\leqslant l-1}, \mathbf{z}^{k+1}_{\leqslant l-1}, \mathbf{a}^{k+1}_{\leqslant l-1}\right)$, then W^{k+1}_l is updated again by setting $\overline{W}^{k+1}_l = W^k_l$ in Line 8 of Algorithm 1, which ensures the decline of F. The same procedure is applied in Lines 13–15, Lines 18–20, and Lines 22–24 in Algorithm 1, respectively.

Next, all subproblems are shown as follows:

1. Update W_l

The variables $W_l(l = 1, \cdots, L)$ are updated as follows:

$$W^{k+1}_l \leftarrow \arg\min_{W_l} \phi(a^{k+1}_{l-1}, W_l, z^k_l) + \Omega_l(W_l). \tag{1}$$

Because W_l and a_{l-1} are coupled in $\phi(\bullet)$, solving W_l requires an inversion operation of a^{k+1}_{l-1}, which is computationally expensive. Motivated by the dlADMM algorithm [31], we define $P^{k+1}_l\left(W_l; \theta^{k+1}_l\right)$ as a quadratic approximation of ϕ at W^k_l as follows:

$$P^{k+1}_l\left(W_l; \theta^{k+1}_l\right) = \phi\left(a^{k+1}_{l-1}, \overline{W}^{k+1}_l, z^k_l\right) + \left(\nabla_{\overline{W}^{k+1}_l}\phi\right)^T \left(W_l - \overline{W}^{k+1}_l\right) + \frac{\theta^{k+1}_l}{2}\|W_l - \overline{W}^{k+1}_l\|^2_2,$$

where $\theta^{k+1}_l > 0$ is a scalar parameter, which can be chosen by the backtracking algorithm [31] to meet the following condition

$$P^{k+1}_l\left(W^{k+1}_l; \theta^{k+1}_l\right) \geqslant \phi\left(a^{k+1}_{l-1}, W^{k+1}_l, z^k_l\right). \tag{2}$$

Rather than minimizing Eq. (1), we instead minimize the following:

$$W^{k+1}_l \leftarrow \arg\min_{W_l} P^{k+1}_l\left(W_l; \theta^{k+1}_l\right) + \Omega_l(W_l). \tag{3}$$

For $\Omega_l(W_l)$, common regularization terms like ℓ_1 or ℓ_2 regularizations lead to closed-form solutions.

2. Update z_l

The variables $z_l(l = 1, \cdots, L)$ are updated as follows:

$$z^{k+1}_l \leftarrow \arg\min_{z_l} \phi\left(a^{k+1}_{l-1}, W^{k+1}_l, z_l\right), \quad s.t.\ h_l(z_l) - \varepsilon \leqslant a_l \leqslant h_l(z_l) + \varepsilon\ (l < L),$$

$$z^{k+1}_L \leftarrow \arg\min_{z_L} \phi\left(a^{k+1}_{L-1}, W^{k+1}_L, z_L\right) + R(z_L; y).$$

Similar to updating W_l, we define $V^{k+1}_l(z_l)$ as follows:

$$V^{k+1}_l(z_l) = \phi\left(a^{k+1}_{l-1}, W^{k+1}_l, \bar{z}^{k+1}_l\right) + \left(\nabla_{\bar{z}^{k+1}_l}\phi\right)^T \left(z_l - \bar{z}^{k+1}_l\right) + \frac{\rho}{2}\|z_l - \bar{z}^{k+1}_l\|^2_2.$$

Hence, we solve the following problems:

$$z^{k+1}_l \leftarrow \arg\min_{z_l} V^{k+1}_l(z_l),\ s.t.\ h_l(z_l) - \varepsilon \leqslant a_l \leqslant h_l(z_l) + \varepsilon\ (l < L). \tag{4}$$

$$z^{k+1}_L \leftarrow \arg\min_{z_L} V^{k+1}_L(z_L) + R(z_L; y). \tag{5}$$

As for $z_l(l = 1, \cdots, l-1)$, the solution is

$$z^{k+1}_l \leftarrow \min\left(\max\left(B^{k+1}_1, \bar{z}^{k+1}_l - \nabla\phi_{\bar{z}^{k+1}_l}/\rho\right), B^{k+1}_2\right),$$

where B^{k+1}_1 and B^{k+1}_2 represent the lower bound and the upper bound of the set $\{z_l | h_l(z_l) - \varepsilon \leqslant a^k_l \leqslant h_l(z_l) + \varepsilon\}$. Eq. (5) is easy to solve using the Fast Iterative Soft Thresholding Algorithm (FISTA) [2].

3. Update a_l

The variables $a_l(l = 1, \cdots, L-1)$ are updated as follows:

$$a^{k+1}_l \leftarrow \arg\min_{a_l} \phi\left(a_l, W^k_{l+1}, z^k_{l+1}\right),\ s.t.\ h_l(z^{k+1}_l) - \varepsilon \leqslant a_l \leqslant h_l(z^{k+1}_l) + \varepsilon.$$

Similar to updating W^{k+1}_l, $Q^{k+1}_l(a_l; \tau^{k+1}_l)$ is defined as

$$Q^{k+1}_l(a_l; \tau^{k+1}_l) = \phi\left(\bar{a}^{k+1}_l, W^k_{l+1}, z^k_{l+1}\right) + \left(\nabla_{\bar{a}^{k+1}_l}\phi\right)^T \left(a_l - \bar{a}^{k+1}_l\right) + \frac{\tau^{k+1}_l}{2}\|a_l - \bar{a}^{k+1}_l\|^2_2,$$

and this allows us to solve the following problem instead:

$$a^{k+1}_l \leftarrow \arg\min_{a_l} Q^{k+1}_l(a_l; \tau^{k+1}_l),\ s.t.\ h_l(z^{k+1}_l) - \varepsilon \leqslant a_l \leqslant h_l(z^{k+1}_l) + \varepsilon, \tag{6}$$

where $\tau^{k+1}_l > 0$ is a scalar parameter, which can be chosen by the backtracking algorithm [31] to meet the following condition:

$$Q^{k+1}_l(a^{k+1}_l; \tau^{k+1}_l) \geqslant \phi\left(a^{k+1}_l, W^k_{l+1}, z^k_{l+1}\right).$$

The solution can be obtained by

$$a^{k+1}_l \leftarrow \min\left(\max\left(h_l(z^{k+1}_l) - \varepsilon, \bar{a}^{k+1}_l - \nabla_{\bar{a}^{k+1}_l}\phi/\tau^{k+1}_l\right), h_l(z^{k+1}_l) + \varepsilon\right).$$

4. Convergence analysis

In this section, the convergence of the proposed algorithm is analyzed. Due to space limit, all proofs are detailed in the appendix. The following mild assumption is required for the convergence analysis of the proposed mDLAM algorithm:

Assumption 2. $F(\mathbf{W}, \mathbf{z}, \mathbf{a})$ is coercive over the domain $\{(\mathbf{W}, \mathbf{z}, \mathbf{a}) | h_l(z_l) - \varepsilon \leqslant a_l \leqslant h_l(z_l) + \varepsilon\ (l = 1, \cdots, L-1)\}$.

The coercivity is defined in the Appendix. Assumption 2 is also mild such that common loss functions such as the least square loss and the cross-entropy loss satisfy it [31].

4.1. Convergence properties

Firstly, the following preliminary lemma is useful to prove the convergence properties of the proposed mDLAM algorithm.

Lemma 1. In Algorithm 1, there exist $\alpha^k_l, \gamma^k_l, \delta^k_l > 0$ such that for $\forall k \in \mathbb{N}, W^k_l, z^k_l(l = 1, 2, \cdots, L)$, and $a^k_l(l = 1, 2, \cdots, L-1)$, it holds that

$$F\left(\mathbf{W}^{k+1}_{\leqslant l-1}, \mathbf{z}^{k+1}_{\leqslant l-1}, \mathbf{a}^{k+1}_{\leqslant l-1}\right) - F\left(\mathbf{W}^{k+1}_{\leqslant l}, \mathbf{z}^{k+1}_{\leqslant l-1}, \mathbf{a}^{k+1}_{\leqslant l-1}\right) \geqslant \frac{\alpha^{k+1}_l}{2}\|W^{k+1}_l - W^k_l\|^2_2. \tag{7}$$

$$F\left(\mathbf{W}^{k+1}_{\leqslant l}, \mathbf{z}^{k+1}_{\leqslant l-1}, \mathbf{a}^{k+1}_{\leqslant l-1}\right) - F\left(\mathbf{W}^{k+1}_{\leqslant l}, \mathbf{z}^{k+1}_{\leqslant l}, \mathbf{a}^{k+1}_{\leqslant l-1}\right) \geqslant \frac{\gamma^{k+1}_l}{2}\|z^{k+1}_l - z^k_l\|^2_2. \tag{8}$$

$$F\left(\mathbf{W}^{k+1}_{\leqslant l}, \mathbf{z}^{k+1}_{\leqslant l}, \mathbf{a}^{k+1}_{\leqslant l-1}\right) - F\left(\mathbf{W}^{k+1}_{\leqslant l}, \mathbf{z}^{k+1}_{\leqslant l}, \mathbf{a}^{k+1}_{\leqslant l}\right) \geqslant \frac{\delta^{k+1}_l}{2}\|a^{k+1}_l - a^k_l\|^2_2. \tag{9}$$

It shows that the objective decreases when all variables are updated. Based on Assumption 2 and Lemma 1, three convergence properties hold, which are shown in the following:

Lemma 2 (*Objective Decrease*). In Algorithm 1, it holds that for any $k \in \mathbb{N}, F\left(\mathbf{W}^k, \mathbf{z}^k, \mathbf{a}^k\right) \geqslant F\left(\mathbf{W}^{k+1}, \mathbf{z}^{k+1}, \mathbf{a}^{k+1}\right)$. Moreover, F is convergent. That is, $F\left(\mathbf{W}^k, \mathbf{z}^k, \mathbf{a}^k\right) \to F^*$ as $k \to \infty$, where F^* is the convergent value of F.

This lemma guarantees the decrease and hence convergence of the objective.

Lemma 3. [Bounded Objective and Variables] In Algorithm 1, it holds that for any $k \in \mathbb{N}$

(a). $\mathbf{F}\left(\mathbf{W}^k, \mathbf{z}^k, \mathbf{a}^k\right)$ is upper bounded. Moreover, $\lim_{k\to\infty} \mathbf{W}^{k+1} - \mathbf{W}^k = 0, \lim_{k\to\infty} \mathbf{z}^{k+1} - \mathbf{z}^k = 0$, and $\lim_{k\to\infty} \mathbf{a}^{k+1} - \mathbf{a}^k = 0$.
(b). $\left(\mathbf{W}^k, \mathbf{z}^k, \mathbf{a}^k\right)$ is bounded. That is, there exist scalars $M_\mathbf{W}, M_\mathbf{z}$ and $M_\mathbf{a}$ such that $\|\mathbf{W}^k\| \leqslant M_\mathbf{W}, \|\mathbf{z}^k\| \leqslant M_\mathbf{z}$ and $\|\mathbf{a}^k\| \leqslant M_\mathbf{a}$.

This lemma ensures that the objective and all variables are bounded in the proposed mDLAM algorithm. Moreover, the gap between the same variables in the neighboring iterations (e.g. $\mathbf{W}^{k+1}$ and $\mathbf{W}^k$) is convergent to 0.

Lemma 4. [Subgradient Bound] In Algorithm 1, there exist $C_2 = \max\left(\rho M_\mathbf{a}, \rho M_\mathbf{a}^2 + \theta_1^{k+1}, \rho M_\mathbf{a}^2 + \theta_2^{k+1}, \cdots, \rho M_\mathbf{a}^2 + \theta_L^{k+1}\right)$, and $g_1^{k+1} \in \partial_{\mathbf{W}^{k+1}} F$ such that for any $k \in \mathbb{N}$

$$\|g_1^{k+1}\| \leqslant C_2\left(\|\mathbf{W}^{k+1} - \mathbf{W}^k\| + \|\mathbf{z}^{k+1} - \mathbf{z}^k\| + \|\mathbf{W}^k - \mathbf{W}^{k-1}\|\right).$$

The above lemma states that the subgradient of the objective is bounded by its variables. This suggests that the subgradient is convergent to 0, and thus proves its convergence to a stationary point.

4.2. Convergence of the proposed mDLAM algorithm

Next we discuss the convergence of the proposed mDLAM algorithm. The first theorem guarantees that the proposed mDLAM algorithm converges to a stationary point.

Theorem 1. [Convergence to a Stationary Point] In Algorithm 1, for $\mathbf{W}$ in Problem 2, for any $\rho > 0$ and $\varepsilon > 0$, starting from any $\mathbf{W}^0$, any limit point $\mathbf{W}^*$ is a stationary point of Problem 2. That is, $0 \in \partial_{\mathbf{W}^*} F$.

As stated in Theorem 1, the convergence always holds no matter how $\mathbf{W}$ is initialized, and whatever ρ and ε are chosen. It is better than the dlADMM algorithm [31], which requires the hyperparameter to be sufficiently large.

Theorem 2. [Linear Convergence Rate] In Algorithm 1, if F is locally strongly convex, then for any ρ, there exist $\varepsilon > 0, k_1 \in \mathbb{N}$ and $0 < C_1 < 1$ such that it holds for $k > k_1$ that

$$F\left(\mathbf{W}^{k+1}, \mathbf{z}^{k+1}, \mathbf{a}^{k+1}\right) - F^* \leqslant C_1\left(F\left(\mathbf{W}^{k-1}, \mathbf{z}^{k-1}, \mathbf{a}^{k-1}\right) - F^*\right).$$

Theorem 2 shows that the proposed mDLAM algorithm converges linearly for sufficiently large iterations. Common loss functions like the square loss or the cross-entropy loss are locally strongly convex [34], which make F locally strongly convex. Therefore, Theorem 2 covers a wide range of loss functions. Compared with existing alternating minimization methods (e.g. dlADMM [31]) with a sublinear $o(1/k)$ convergence rate, the proposed mDLAM algorithm achieves a theoretically better linear convergence rate.

4.3. Discussion

We discuss convergence conditions of the proposed mDLAM algorithm compared with SGD-type methods and the dlADMM method. The comparison demonstrates that our convergence conditions are more general than others.

1. mDLAM versus SGD

One influential work by Ghadimi et al. [9] guaranteed that the SGD converges to a critical point, which is similar to our convergence results. While the SGD requires the objective function to be Lipschitz differentiable, bounded from below [9], our mDLAM allows for non-smooth functions such as ReLU. Therefore, our convergence conditions are milder than SGD.

2. mDLAM versus dlADMM

Wang et al. [31] proposed an improved version of ADMM for deep learning models called dlADMM. They showed that the dlADMM is convergent to a critical point. However, assumptions of our mDLAM are milder than those of the dlADMM: the mDLAM requires activation functions to be quasilinear, which includes sigmoid, tanh, ReLU, and leaky ReLU, while the dlADMM assumes that activation functions make subproblems solvable, which only includes ReLU and leaky ReLU. Such difference originates from different ways of addressing nonlinear activations: the dlADMM treats them as L_2 penalties. For tanh and sigmoid, subproblems are difficult to solve and may refer to lookup tables [31]. However, the mDLAM relaxes them via inequality constraints, and subproblems have closed-form solutions.

5. Experiments

In this section, we evaluate the proposed mDLAM algorithm on four benchmark datasets. Convergence and efficiency are demonstrated. The performance of the proposed mDLAM algorithm is compared with several state-of-the-art optimizers. All experiments were conducted on a 64-bit machine with Intel(R) Xeon(R) Silver 4110 CPU and 64 GB RAM.

5.1. Datasets and parameter settings

An important application of the MLP model is node classification on a graph based on augmented node features [5]. Specifically, given an adjacency matrix A and a node feature matrix H of a graph, we let the k-th augmented feature $X^k = HA^k (k = 0, 1, \cdots, 4)$, which encodes information of graph topology via A^k, and then concatenate them into the input $X = [X_0, \cdots, X_4]$ [5]. The MLP model is used to predict the node class based on the input X. We set up an architecture of three layers, each of which has 100 hidden units. The activation function was set to ReLU. The number of epoch was set to 200. We test our model on four benchmark datasets: Cora [23], Pubmed [23], Citeseer [23] and Coauthor CS [24], whose statistics are shown in Table 2.

Gradient Descent (GD) [3], Adaptive learning rate method (Adadelta) [36], Adaptive gradient algorithm (Adagrad) [8], Adaptive momentum estimation (Adam) [12], and deep learning Alternating Direction Method of Multipliers (dlADMM) [31] are state-of-the-art methods and hence were served as comparison methods. The full batch dataset was used for training models. All parameters were chosen by maximizing the accuracy of training datasets. Table 3 shows hyperparameters of all methods: for the proposed mDLAM algorithm, ρ controls quadratic terms in Problem 2; α is a learning rate in the comparison methods except for dlADMM. ρ controls a linear constraint in the dlADMM algorithm. The other hyperparameter ε is chosen adaptively as follows: $\varepsilon^{k+1} = \max(\varepsilon^k/2, 0.001)$ with $\varepsilon^0 = 100$. This makes inequality constraints relaxed at the early stage (i.e. ε^k is large and hence constraints are easy to satisfy) and then tightens them as the mDLAM iterates.

5.2. Convergence

Firstly, we investigate the convergence of the proposed mDLAM algorithm on four benchmark datasets using the hyperparameters summarized in Table 3. The relationship between the objective and the number of epochs is shown in Fig. 1. Overall, the objectives on the four datasets all decrease monotonically, which demonstrates the convergence of the proposed mDLAM algorithm. Nevertheless, objective curves vary in tendency: the curves on the Cora and Pubmed datasets drop drastically at the beginning and then reach the plateau when the epoch is around 75, while the curves on the other two datasets keep a downward tendency in the entire 200 epochs. Moreover, the objective on the Pubmed dataset is the lowest at the end of the training, while the objective on the Citeseer dataset is in the vicinity of 80, at least 60% higher than objectives on the remaining datasets. It is easy to observe that all curves decline linearly when the epoch is higher than 100. This validates the linear convergence rate of our proposed mDLAM algorithm (i.e. Theorem 2).

5.3. Performance

Next, the performance of the proposed mDLAM algorithm is compared against five state-of-the-art methods, as is illustrated in Fig. 2. X-axis and Y-axis represent epoch and test accuracy, respectively. Overall, the proposed mDLAM algorithm is superior to all other algorithms on four datasets, which has not only the highest test accuracy but also the fastest convergence speed. For example, the proposed mDLAM achieves 70% test accuracy on the Cora dataset when the epoch is 100, while GD only attains 60%, and the Adadelta reaches the plateau of around 40%; As another example, the test accuracy of the proposed mDLAM on the Coauthor CS dataset is over 80% at the 25-th epoch, whereas most comparison methods such as Adam and GD reach half of its accuracy (i.e. 40%). The Adadelta algorithm performs the worst among all comparison methods: it converges to a low test accuracy at the early stage, which is usually half of the accuracy accomplished by the proposed mDLAM algorithm. The other four comparison methods except Adagrad are on par with mDLAM in some cases: for example, the curves of dlADMM and GD are marginally behind that of mDLAM on the Coauthor CS dataset, and the performance of Adam almost reaches that of mDLAM on the Cora dataset. It is interesting to observe that curves of some methods decline at the end of 200 epochs such as the Adagrad on the Pubmed dataset and the Adam on the Coauthor CS dataset.

5.4. Sensitivity analysis

We explore concerning factors of the running time and the test accuracy in this section.

5.4.1. Running time

Moreover, it is important to explore the running time of the proposed mDLAM concerning two factors: the number of hidden units and the value of ρ. The running time was averaged by 200 epochs. Fig. 3(a) depicts the relationship between the running time and the number of hidden units on four datasets, where the number of hidden units ranges from 100 to 1,000. The running times on all datasets are below 1 s per epoch when the number of hidden units is 100, and increase linearly with the number of hidden units in general. However, the rates of increase vary on different datasets: the curve on the Coauthor CS dataset has the sharpest slope, which reaches seven seconds per epoch when 1000 hidden units are applied, while the curve on the Pubmed dataset climbs slowly, which never surpasses 1 s. The curves on the Cora and the Citeseer datasets demonstrate a steady increase.

To investigate the relationship between the running time per epoch and the value of ρ, we change ρ from 10^{-6} to 1 while fixing others. Similar to Fig. 3(a), the running time per epoch demonstrates a linear increase concerning the value of ρ in general, as shown in Fig. 3(b). Specifically, the curve on the Coauthor CS dataset is still the highest in slope, whereas the slope on the Pubmed dataset is the lowest. Moreover, the effect of the value of ρ is less obvious than the number of hidden units. For example, in Fig. 3(b) when ρ is enlarged from 10^{-6} to 10^{-2}, the running time on the Coauthor CS dataset merely ascends from around 0.65 to 1.6, while the increment of the running time on other datasets is less than 0.2. Moreover, a larger ρ may reduce the running time. For

Table 2
Statistics of four benchmark datasets.

Dataset	Node#	Edge#	Class#	Feature#
Cora	2708	5429	7	1433
Pubmed	19717	44338	3	500
Citeseer	3327	4732	6	3703
Coauthor CS	18333	81894	15	6805

Table 3
Hyperparameter settings on four datasets: they were chosen based on training performance.

Method	Hyper- parameters	Cora	Pubmed	Citeseer	Coauthor CS
mDLAM	ρ	1×10^{-3}	0.01	5×10^{-3}	1×10^{-4}
GD	α	0.01	0.01	0.01	5×10^{-3}
Adadelta	α	0.01	0.1	0.01	0.05
Adagrad	α	5×10^{-3}	5×10^{-3}	0.01	5×10^{-3}
Adam	α	1×10^{-3}	5×10^{-4}	1×10^{-3}	1×10^{-3}
dlADMM	ρ	1×10^{-6}	1×10^{-6}	1×10^{-6}	1×10^{-6}

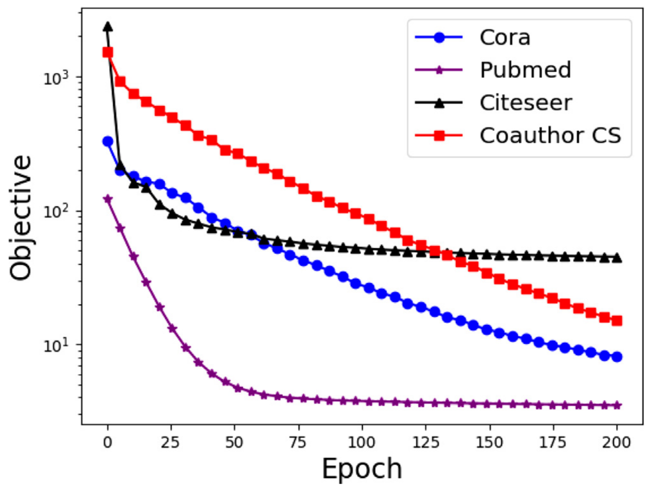

Fig. 1. Convergence curves on four datasets: they all converge linearly when the epoch is larger than 100.

instance, when ρ increases from 10^{-2} to 1, the running time on the Coauthor CS dataset drops slightly from 1.6 s to 1.5 s per epoch. The running times on the Cora and the Citeseer datasets climb steadily.

5.4.2. Test accuracy

Finally, we investigate the effects of hyperparameters on test accuracy, namely, the value of ρ and ε. Because ε is dynamically set, we test its initial value ε^0. Table 4 demonstrates the relationship between test accuracy and ρ on four datasets. ρ was chosen from $\left\{1\times10^{-4}, 1\times10^{-3}, 1\times10^{-2}\right\}$. Overall, the choice of ρ has a significant effect on the test accuracy. For example, when ρ is changed from 1×10^{-4} to 1×10^{-3} on the Pubmed dataset, the performance has improved by approximately 60%, and the gain of performance is even roughly 90% if it is modified to 1×10^{-2}. On other datasets, the change of ρ affects test accuracy by around 20%. For instance, the test accuracy on the Cora dataset and the Coauthor CS dataset can be improved to 0.74 and 0.89 if we set $\rho = 1\times10^{-3}$ and $\rho = 1\times10^{-4}$, respectively. The test accuracy on the Citeseer dataset is relatively robust to the change of ρ. As ρ varies from 1×10^{-3} to 1×10^{-2}, the test accuracy remains stable. Obviously, the test accuracy generally increases as the proposed mDLAM algorithm iterates. However, there are some exceptions: for example, the test accuracy has dropped slightly from 0.66 to 0.65 when $\rho = 1\times10^{-3}$ on the Pubmed dataset.

Table 5 shows the relationship between test accuracy and the initial value of ε (i.e. ε^0) on four datasets. ε^0 was chosen from $\{1, 10, 100\}$. It is obvious that test accuracy is resistant to the

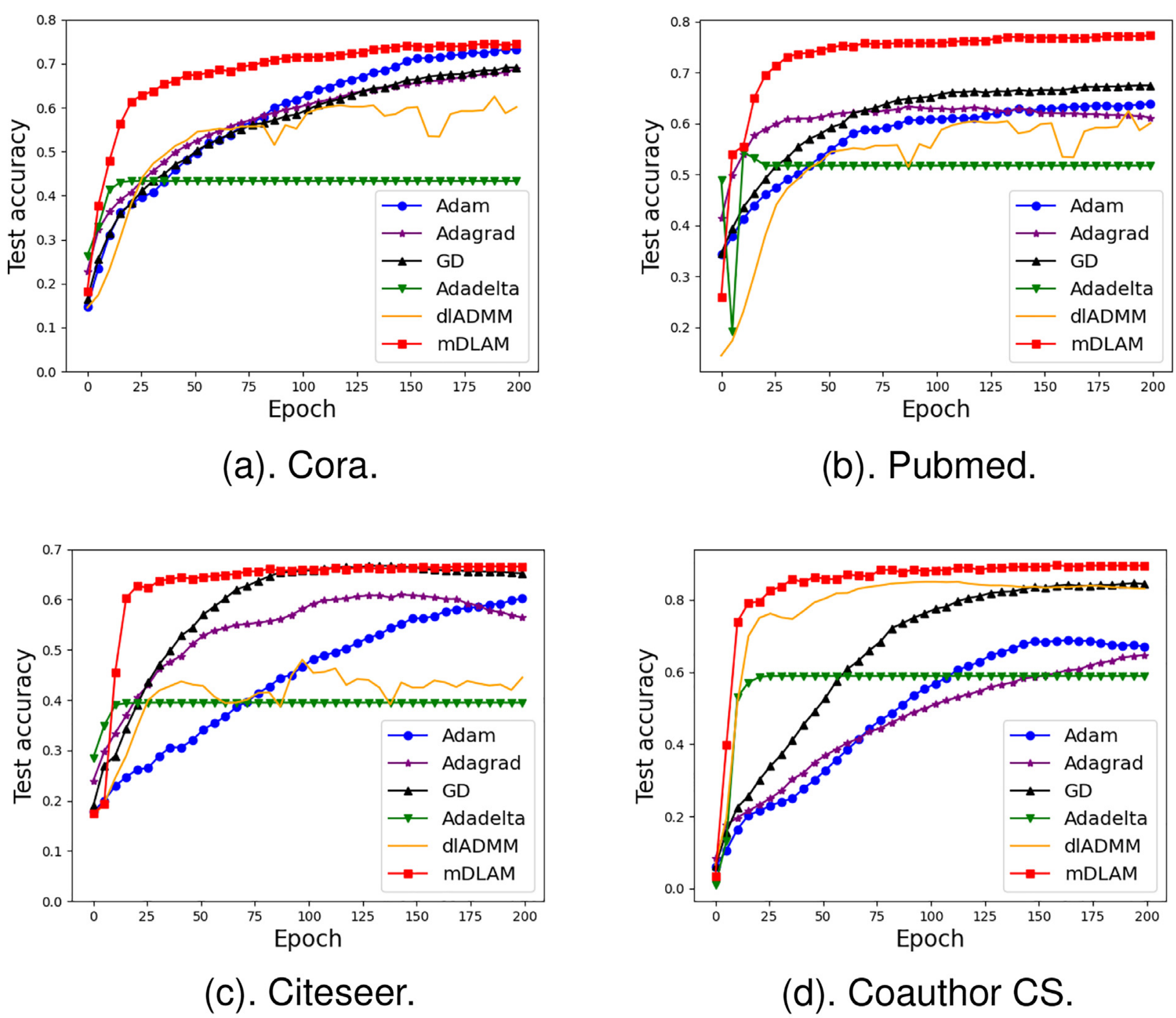

Fig. 2. Test accuracy of all methods: the proposed mDLAM algorithm outperforms all other comparison methods in four datasets.

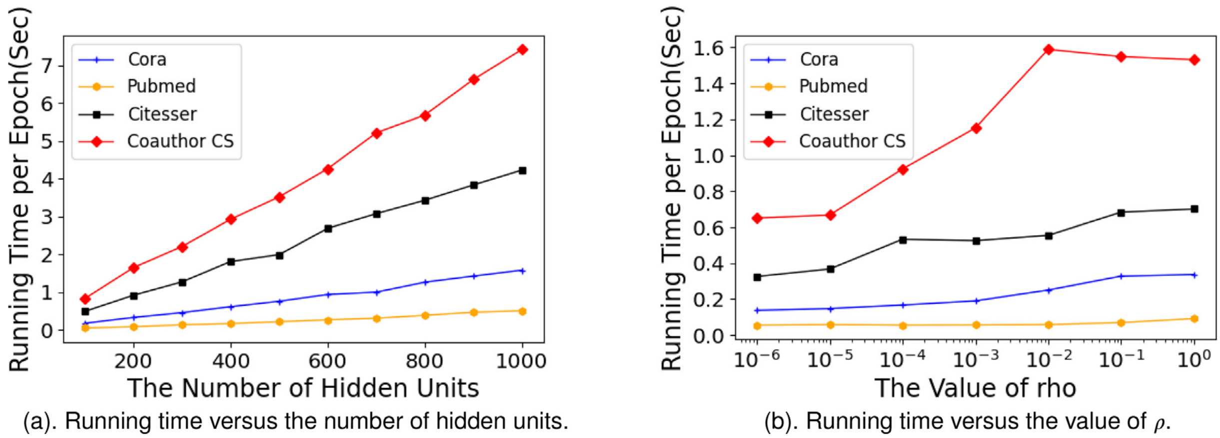

Fig. 3. The relationship between the running time and: (a) the number of hidden units; (b) the value of ρ: the running time increases linearly with them in general.

Table 4
The effect of ρ on test accuracy on four datasets: it affects performance significantly.

			Cora		
Epoch	40	80	120	160	200
$\rho = 1 \times 10^{-4}$	0.677	0.695	0.695	0.693	0.692
$\rho = 1 \times 10^{-3}$	0.664	0.701	0.721	0.737	0.742
$\rho = 1 \times 10^{-2}$	0.562	0.581	0.604	0.623	0.638
			Pubmed		
Epoch	40	80	120	160	200
$\rho = 1 \times 10^{-4}$	0.471	0.407	0.407	0.407	0.407
$\rho = 1 \times 10^{-3}$	0.663	0.645	0.640	0.650	0.649
$\rho = 1 \times 10^{-2}$	0.743	0.758	0.762	0.768	0.773
			Citeseer		
Epoch	40	80	120	160	200
$\rho = 1 \times 10^{-4}$	0.528	0.529	0.530	0.531	0.535
$\rho = 1 \times 10^{-3}$	0.651	0.665	0.664	0.664	0.666
$\rho = 1 \times 10^{-2}$	0.631	0.638	0.642	0.648	0.653
			Coauthor CS		
Epoch	40	80	120	160	200
$\rho = 1 \times 10^{-4}$	0.843	0.881	0.888	0.896	0.894
$\rho = 1 \times 10^{-3}$	0.780	0.807	0.825	0.839	0.835
$\rho = 1 \times 10^{-2}$	0.688	0.719	0.724	0.737	0.738

Table 5
The effect of the initial value of ε (i.e. ε^0) on test accuracy on four datasets: it only affects the convergence speed, but have little effect on final performance.

			Cora		
Epoch	40	80	120	160	200
$\varepsilon^0 = 1$	0.620	0.679	0.712	0.735	0.743
$\varepsilon^0 = 10$	0.646	0.689	0.718	0.741	0.741
$\varepsilon^0 = 100$	0.664	0.701	0.721	0.737	0.742
			Pubmed		
Epoch	40	80	120	160	200
$\varepsilon^0 = 1$	0.717	0.744	0.756	0.759	0.763
$\varepsilon^0 = 10$	0.731	0.753	0.759	0.762	0.765
$\varepsilon^0 = 100$	0.743	0.758	0.762	0.768	0.773
			Citeseer		
Epoch	40	80	120	160	200
$\varepsilon^0 = 1$	0.564	0.615	0.638	0.653	0.663
$\varepsilon^0 = 10$	0.584	0.626	0.643	0.657	0.662
$\varepsilon^0 = 100$	0.640	0.656	0.664	0.663	0.668
			Coauthor CS		
Epoch	40	80	120	160	200
$\varepsilon^0 = 1$	0.834	0.875	0.887	0.893	0.894
$\varepsilon^0 = 10$	0.852	0.866	0.892	0.893	0.893
$\varepsilon^0 = 100$	0.843	0.881	0.888	0.896	0.894

change of ε^0. For example, the test accuracy on the Coauthor CS dataset is in the vicinity of 0.89 no matter what ε is chosen. Moreover, the larger a ε^0 is, the faster convergence speed the proposed mDLAM algorithm gains. For instance, when $\varepsilon = 100$, the test accuracy is 0.08 better than that in the case where $\varepsilon = 1$ on the Citeseer dataset. Compared with Tables 4 and 5, the effect of ρ is more signifcant than that of ε^0.

6. Conclusion

In this paper, we propose a novel formulation of the original neural network problem and a novel monotonous Deep Learning Alternating Minimization (mDLAM) algorithm. Specifically, the nonlinear constraint is projected into a convex set so that all subproblems are solvable. The Nesterov acceleration technique is applied to boost the convergence of the proposed mDLAM algorithm. Furthermore, a mild assumption is established to prove the convergence of our mDLAM algorithm. Our mDLAM algorithm can achieve a linear convergence rate, which is theoretically better than existing alternating minimization methods. The effectiveness of the proposed mDLAM algorithm is demonstrated via the outstanding performance on four benchmark datasets compared with state-of-the-art optimizers.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the National Science Foundation (NSF) Grant No. 1755850, No. 1841520, No. 2007716, No. 2007976, No. 1942594, No. 1907805, a Jeffress Memorial Trust Award, Amazon Research Award, NVIDIA GPU Grant, and Design Knowledge Company (subcontract No: 10827.002.120.04).

Appendix A. Definition

Several definitions are shown here for the sake of convergence analysis.

Definition 1 (*Coercivity*). Any arbitrary function $G_2(x)$ is coercive over a nonempty set $dom(G_2)$ if as $\|x\| \to \infty$ and $x \in dom(G_2)$, we have $G_2(x) \to \infty$, where $dom(G_2)$ is a domain set of G_2.

Definition 2 (*Multi-convexity*). A function $f(x_1, x_2, \cdots, x_m)$ is a multi-convex function if f is convex with regard to $x_i (i = 1, \cdots, m)$ while fixing other variables.

Definition 3 (*Lipschitz Differentiability*). A function $f(x)$ is Lipschitz differentiable with Lipschitz coefficient $L > 0$ if for any $x_1, x_2 \in \mathbb{R}$, the following inequality holds:

$$\|\nabla f(x_1) - \nabla f(x_2)\| \leqslant L\|x_1 - x_2\|.$$

For Lipschitz differentiability, we have the following lemma (Lemma 2.1 in [2]):

Lemma 5. If $f(x)$ is Lipschitz differentiable with $L > 0$, then for any $x_1, x_2 \in \mathbb{R}$

$$f(x_1) \leqslant f(x_2) + \nabla f^T(x_2)(x_1 - x_2) + \frac{L}{2}\|x_1 - x_2\|^2.$$

Definition 4 (*Fréchet Subdifferential*). For each $x_1 \in dom(u_1)$, the Fréchet subdifferential of u_1 at x_1, which is denoted as $\hat{\partial} u_1(x_1)$, is the set of vectors v, which satisfy

$$\liminf_{x_2 \neq x_1 x_2 \to x_1} \left(u_1(x_2) - u_1(x_1) - v^T(x_2 - x_1)\right)/\|x_2 - x_1\| \geqslant 0.$$

The vector $v \in \hat{\partial} u_1(x_1)$ is a Fréchet subgradient.

Then the definition of the limiting subdifferential, which is based on Fréchet subdifferential, is given in the following [21]:

Definition 5 (*Limiting Subdifferential*). For each $x \in dom(u_2)$, the limiting subdifferential (or subdifferential) of u_2 at x is

$$\partial u_2(x) = \left\{ v_1 | \exists\, x^k \to x, s.t.\ u_2(x^k) \to u_2(x), v^k \in \hat{\partial} u_2(x^k), v^k \to v \right\},$$

where x^k is a sequence whose limit is x and the limit of $u_2(x^k)$ is $u_2(x)$, v^k is a sequence, which is a Fréchet subgradient of u_2 at x^k and whose limit is v. The vector $v \in \partial u_2(x)$ is a limiting subgradient.

Specifically, when u_2 is convex, its limiting subdifferential is reduced to regular subdifferential [21], which is defined as follows:

Definition 6 (*Regular Subdifferential*). For each $x_1 \in dom(f)$, the regular subdifferential of a convex function f at x_1, which is denoted as $\partial f(x_1)$, is the set of vectors v, which satisfy

$$f(x_2) \geqslant f(x_1) + v^T(x_2 - x_1).$$

The vector $v \in \partial f(x_1)$ is a regular subgradient.

Definition 7 (*Quasilinearity*). A function $f(x)$ is quasiconvex if for any sublevel set $S_v(f) = \{x | f(x) \leqslant v\}$ is a convex set. Likewise, A function $f(x)$ is quasiconcave if for any superlevel set $S_v(f) = \{x | f(x) \geqslant v\}$ is a convex set. A function $f(x)$ is quasilinear if it is both quasiconvex and quasiconcave.

Definition 8 (*Locally Strong Convexity*). A function $f(x)$ is locally strongly convex within a bound set $\mathbb{D}$ with a constant μ if

$$f(y) \geqslant f(x) + g^T(y - x) + \frac{\mu}{2}\|x - y\|_2^2 \ \forall\ g \in \partial f(x) \text{and } x, y \in \mathbb{D}.$$

Simply speaking, a locally strongly convex function lies above a quadratic function within a bounded set.

Definition 9 (*Kurdyka-Lojasiewicz (KL) Property*). A function $f(x)$ has the KL Property at $\bar{x} \in dom\ \partial f = \{x \in \mathbb{R} : \partial f(x) \neq \emptyset\}$ if there exists $\eta \in (0, +\infty]$, a neighborhood X of $\bar{x}$ and a function $\psi \in \Psi_\eta$, such that for all

$$x \in X \cap \{x \in \mathbb{R} : f(\bar{x}) < f(x) < f(\bar{x}) + \eta\},$$

the following inequality holds

$$\psi'(f(x) - f(\bar{x})) dist(0, \partial f(x)) \geqslant 1,$$

where Ψ_η stands for a class of function $\psi : [0, \eta] \to \mathbb{R}^+$ satisfying: (1). ϕ is concave and $\psi'(x)$ continuous on $(0, \eta)$; (2). ψ is continuous at 0, $\psi(0) = 0$; and (3). $\psi'(x) > 0, \forall x \in (0, \eta)$.

The following lemma shows that a locally strongly convex function satisfies the KL Property:

Lemma 6 [34]. A locally strongly convex function $f(x)$ with a constant μ satisfies the KL Property at any $x \in \mathbb{D}$ with $\psi(x) = \frac{2}{\mu}\sqrt{x}$ and $X = \mathbb{D} \cap \{y : f(y) \geqslant f(x)\}$.

Appendix B. Preliminary results

In this section, we present preliminary lemmas of the proposed mDLAM algorithm. The limiting subdifferential is used to prove the convergence of the proposed mDLAM algorithm in the following convergence analysis. Without loss of generality, ∂R and $\partial \Omega_l (l = 1, \cdots, n)$ are assumed to be nonempty, and the limiting subdifferential of F defined in Problem 2 is [34]:

$\partial F(\mathbf{W}, \mathbf{z}, \mathbf{a}) = \partial_{\mathbf{W}} F \times \partial_{\mathbf{z}} F \times \partial_{\mathbf{a}} F,$

where $\times$ means the Cartesian product.

Lemma 7. holds, then there exists $p \in \partial \Omega_l \left(W_l^{k+1}\right)$, the subgradient of $\Omega_l\left(W_l^{k+1}\right)$ such that

$$\nabla_{\overline{W}_l^{k+1}} \phi + \theta_l^{k+1}\left(W_l^{k+1} - \overline{W}_l^{k+1}\right) + p = 0.$$

Likewise, if Eq. (4) holds, then there exists q such that

$$\nabla_{\overline{z}_l^{k+1}} \phi + \rho\left(z_l^{k+1} - \overline{z}_l^{k+1}\right) + q = 0,$$

where q is a subgradient with regard to z_l^{k+1} to satisfy the constraint $h_l(z_l^{k+1}) - \varepsilon \leqslant a_l^k \leqslant h_l(z_l^{k+1}) + \varepsilon$. If Eq. (5) holds, then there exists $u \in \partial R(z_L^{k+1}; y)$ such that

$$\nabla_{\overline{z}_L^{k+1}} \phi + \rho(z_L^{k+1} - \overline{z}_L^{k+1}) + u = 0.$$

If Eq. (6) holds, then there exists v such that

$$\nabla_{\overline{a}_l^{k+1}} \phi + \tau_l^{k+1}\left(a_l^{k+1} - \overline{a}_l^{k+1}\right) + v = 0,$$

where v is a subgradient with regard to a_l^{k+1} to satisfy the constraint $h_l(z_l^{k+1}) - \varepsilon \leqslant a_l^{k+1} \leqslant h_l(z_l^{k+1}) + \varepsilon$.

Proof. These can be obtained by directly applying the optimality conditions of Eq. (3), Eq. (4), Eq. (5) and Eq. (6), respectively.

Lemma 8. For Eq. (4) and Eq. (5), the following inequalities hold:

$$V_l^{k+1}(z_l^{k+1}) \geqslant \phi\left(a_{l-1}^{k+1}, W_l^{k+1}, z_l^{k+1}\right). \tag{10}$$

Proof. Because $\phi(a_{l-1}, W_l, z_l)$ is Lipschitz differentiable with respect to z_l with Lipschitz coefficient ρ, we directly apply Lemma 5 to ϕ to obtain Eq. (10).

Appendix C. Main proofs

Proof of Lemma 1

Proof. In Algorithm 1, we only show Eq. (7) because Eq. (8) and Eq. (9) follow the same routine of Eq. (7).

In Line 7 of Algorithm 1, if $F\left(\mathbf{W}_{\leqslant l}^{k+1}, \mathbf{z}_{\leqslant l-1}^{k+1}, \mathbf{a}_{\leqslant l-1}^{k+1}\right) < F\left(\mathbf{W}_{\leqslant l-1}^{k+1}, \mathbf{z}_{\leqslant l-1}^{k+1}, \mathbf{a}_{\leqslant l-1}^{k+1}\right)$, then obviously there exists $\alpha_l^{k+1} > 0$ such that Eq. (7) holds. Otherwise, according to Line 8 of Algorithm 1, because $\Omega_{W_l}(W_l)$ and $\phi(a_{l-1}, W_l, z_l)$ are convex with regard to W_l, according to the definition of regular subgradient, we have

$$\Omega_l\left(W_l^k\right) \geqslant \Omega_l\left(W_l^{k+1}\right) + p^T\left(W_l^k - W_l^{k+1}\right), \tag{11}$$

$$\phi\left(a_{l-1}^{k+1}, W_l^k, z_l^k\right) \geqslant \phi(a_{l-1}^{k+1}, \overline{W}_l^{k+1}, z_l^k) + \nabla_{\overline{W}_l^{k+1}} \phi^T\left(W_l^k - \overline{W}_l^{k+1}\right), \tag{12}$$

where p is defined in the premise of Lemma 7. Therefore, we have

$$\begin{aligned}
&F\left(\mathbf{W}_{\leqslant l-1}^{k+1}, \mathbf{z}_{\leqslant l-1}^{k+1}, \mathbf{a}_{\leqslant l-1}^{k+1}\right) - F\left(\mathbf{W}_{\leqslant l}^{k+1}, \mathbf{z}_{\leqslant l-1}^{k+1}, \mathbf{a}_{\leqslant l-1}^{k+1}\right) \\
&= \phi\left(a_{l-1}^{k+1}, W_l^k, z_l^k\right) + \Omega_l\left(W_l^k\right) - \phi\left(a_{l-1}^{k+1}, W_l^{k+1}, z_l^k\right) \\
&-\Omega_l\left(W_l^{k+1}\right) \text{ (Definition of } F \text{ in Problem 2)} \\
&\geqslant \Omega_l\left(W_l^k\right) - \Omega_l\left(W_l^{k+1}\right) - \left(\nabla_{\overline{W}_l^{k+1}} \phi\right)^T\left(W_l^{k+1} - \overline{W}_l^{k+1}\right) \\
&-\frac{\theta_l^{k+1}}{2}\|W_l^{k+1} - \overline{W}_l^{k+1}\|_2^2 - \phi\left(a_{l-1}^{k+1}, \overline{W}_l^{k+1}, z_l^k\right) \\
&\qquad +\phi\left(a_{l-1}^{k+1}, W_l^k, z_l^k\right) \text{(Eq. (2))} \\
&\geqslant p^T\left(W_l^k - W_l^{k+1}\right) - \left(\nabla_{\overline{W}_l^{k+1}} \phi\right)^T\left(W_l^{k+1} - W_l^k\right) \\
&-\frac{\theta_l^{k+1}}{2}\|W_l^{k+1} - \overline{W}_l^{k+1}\|_2^2 \text{ (Eq. (11) and Eq. (12))} \\
&= -\left(\nabla_{\overline{W}_l^{k+1}} \phi + \theta_l^{k+1}\left(W_l^{k+1} - \overline{W}_l^{k+1}\right)\right)^T\left(W_l^k - W_l^{k+1}\right) \\
&-\left(\nabla_{\overline{W}_l^{k+1}} \phi\right)^T\left(W_l^{k+1} - W_l^k\right) - \frac{\theta_l^{k+1}}{2}\|W_l^{k+1} - \overline{W}_l^{k+1}\|_2^2 \text{(Lemma 7)} \\
&= \frac{\theta_l^{k+1}}{2}\|W_l^{k+1} - \overline{W}_l^{k+1}\|_2^2 + \theta_l^{k+1}\left(W_l^{k+1} - \overline{W}_l^{k+1}\right)^T\left(\overline{W}_l^{k+1} - W_l^k\right) \\
&= \frac{\theta_l^{k+1}}{2}\left(\|W_l^{k+1} - W_l^k\|_2^2 - \|\overline{W}_l^{k+1} - W_l^k\|_2^2\right) \\
&\qquad = \frac{\theta_l^{k+1}}{2}\|W_l^{k+1} - W_l^k\|_2^2 \left(\overline{W}_l^{k+1} = W_l^k\right).
\end{aligned}$$

Let $\alpha_l^{k+1} = \theta_l^{k+1}$, then Eq. (7) still holds.

Proof of Lemma 3

Proof. In Algorithm 1:(a). We sum Eq. (7), Eq. (8) and Eq. (9) from $l = 1$ to L and from $k = 0$ to K to obtain

$$\begin{aligned}
&F\left(\mathbf{W}^0, \mathbf{z}^0, \mathbf{a}^0\right) - F\left(\mathbf{W}^K, \mathbf{z}^K, \mathbf{a}^K\right) \\
&\geqslant \sum_{k=0}^{K}\left(\sum_{l=1}^{L}\left(\frac{\alpha_l^{k+1}}{2}\|W_l^{k+1} - W_l^k\|_2^2 + \frac{\gamma_l^{k+1}}{2}\|z_l^{k+1} - z_l^k\|_2^2\right)\right. \\
&\left. + \sum_{l=1}^{L-1}\frac{\delta_l^{k+1}}{2}\|a_l^{k+1} - a_l^k\|_2^2\right).
\end{aligned} \tag{13}$$

So $F\left(\mathbf{W}^K, \mathbf{z}^K, \mathbf{a}^K\right) \leqslant F\left(\mathbf{W}^0, \mathbf{z}^0, \mathbf{a}^0\right)$. This proves the upper boundness of F. Let $K \to \infty$ in Eq. (13), since $F > 0$ is lower bounded, we have

$$\begin{aligned}
&\sum_{k=0}^{K}\left(\sum_{l=1}^{L}\left(\frac{\alpha_l^{k+1}}{2}\|W_l^{k+1} - W_l^k\|_2^2 + \frac{\gamma_l^{k+1}}{2}\|z_l^{k+1} - z_l^k\|_2^2\right)\right. \\
&\left. + \sum_{l=1}^{L-1}\frac{\delta_l^{k+1}}{2}\|a_l^{k+1} - a_l^k\|_2^2\right) < \infty.
\end{aligned} \tag{14}$$

Since the sum of this infinite series is finite, every term converges to 0. This means that $\lim_{k\to\infty} W_l^{k+1} - W_l^k = 0, \lim_{k\to\infty} z_l^{k+1} - z_l^k = 0$ and $\lim_{k\to\infty} a_l^{k+1} - a_l^k = 0$. In other words, $\lim_{k\to\infty} \mathbf{W}^{k+1} - \mathbf{W}^k = 0$, $\lim_{k\to\infty} \mathbf{z}^{k+1} - \mathbf{z}^k = 0$, and $\lim_{k\to\infty} \mathbf{a}^{k+1} - \mathbf{a}^k = 0$.(b). Because $F\left(\mathbf{W}^k, \mathbf{z}^k, \mathbf{a}^k\right)$ is bounded, by the definition of coercivity and Assumption 2, $\left(\mathbf{W}^k, \mathbf{z}^k, \mathbf{a}^k\right)$ is bounded.

Proof of Lemma 4

Proof. As shown in Remark 2.2 in [34],

$$\partial_{\mathbf{W}^{k+1}} F = \left\{\partial_{W_1^{k+1}} F\right\} \times \left\{\partial_{W_2^{k+1}} F\right\} \times \cdots \times \left\{\partial_{W_L^{k+1}} F\right\},$$

where $\times$ denotes Cartesian Product.

In Algorithm 1, for W_l^{k+1}, according to Line 6 of Algorithm 1, if. $F\left(\mathbf{W}_{\leqslant l}^{k+1}, \mathbf{z}_{\leqslant l-1}^{k+1}, \mathbf{a}_{\leqslant l-1}^{k+1}\right) < F\left(\mathbf{W}_{\leqslant l-1}^{k+1}, \mathbf{z}_{\leqslant l-1}^{k+1}, \mathbf{a}_{\leqslant l-1}^{k+1}\right)$, then

$$\begin{aligned}
&\partial_{W_l^{k+1}} F = \partial\Omega_l\left(W_l^{k+1}\right) + \nabla_{W_l^{k+1}}\phi\left(a_{l-1}^{k+1}, W_l^{k+1}, z_l^{k+1}\right)\\
&(\text{Definition of } F \text{ in Problem 2})\\
&= \nabla_{W_l^{k+1}}\phi\left(a_{l-1}^{k+1}, W_l^{k+1}, z_l^{k+1}\right) - \nabla_{\overline{W}_l^{k+1}}\phi\left(a_{l-1}^{k+1}, \overline{W}_l^{k+1}, z_l^{k}\right)\\
&-\theta_l^{k+1}\left(W_l^{k+1} - \overline{W}_l^{k+1}\right) + \partial\Omega_l\left(W_l^{k+1}\right)\\
&+\nabla_{\overline{W}_l^{k+1}}\phi\left(a_{l-1}^{k+1}, \overline{W}_l^{k+1}, z_l^{k}\right) + \theta_l^{k+1}\left(W_l^{k+1} - \overline{W}_l^{k+1}\right)\\
&= \rho\left(W_l^{k+1} - \overline{W}_l^{k+1}\right)a_{l-1}^{k+1}(a_{l-1}^{k+1})^T - \rho(z_l^{k+1} - z_l^{k})(a_{l-1}^{k+1})^T\\
&-\theta_l^{k+1}\left(W_l^{k+1} - \overline{W}_l^{k+1}\right) + \partial\Omega_l\left(W_l^{k+1}\right)\\
&\quad+\nabla_{\overline{W}_l^{k+1}}\phi\left(a_{l-1}^{k+1}, \overline{W}_l^{k+1}, z_l^{k}\right) + \theta_l^{k+1}\left(W_l^{k+1} - \overline{W}_l^{k+1}\right).
\end{aligned} \tag{15}$$

On one hand, we have

$$\begin{aligned}
&\|\rho\left(W_l^{k+1} - \overline{W}_l^{k+1}\right)a_{l-1}^{k+1}(a_{l-1}^{k+1})^T - \rho(z_l^{k+1} - z_l^{k})(a_{l-1}^{k+1})^T\\
&-\theta_l^{k+1}\left(W_l^{k+1} - \overline{W}_l^{k+1}\right)\|\\
&\leqslant \rho\|\left(W_l^{k+1} - \overline{W}_l^{k+1}\right)a_{l-1}^{k+1}(a_{l-1}^{k+1})^T\| + \rho\|(z_l^{k+1} - z_l^{k})(a_{l-1}^{k+1})^T\|\\
&+\theta_l^{k+1}\|W_l^{k+1} - \overline{W}_l^{k+1}\|(\text{Triangle Inequality})\\
&\leqslant \rho\|W_l^{k+1} - \overline{W}_l^{k+1}\|\|a_{l-1}^{k+1}\|\|a_{l-1}^{k+1}\| + \rho\|z_l^{k+1} - z_l^{k}\|\|a_{l-1}^{k+1}\| + \theta_l^{k+1}\|W_l^{k+1}\\
&-\overline{W}_l^{k+1}\|(\text{Cauchy} - \text{Schwarz Inequality})\\
&\quad\leqslant \rho M_{\mathbf{a}}\|z_l^{k+1} - z_l^{k}\| + \left(\rho M_{\mathbf{a}}^2 + \theta_l^{k+1}\right)\|W_l^{k+1} - \overline{W}_l^{k+1}\| \ (\text{Lemma 3})\\
&\quad\leqslant \rho M_{\mathbf{a}}\|z_l^{k+1} - z_l^{k}\| + \left(\rho M_{\mathbf{a}}^2 + \theta_l^{k+1}\right)\\
&\quad\|W_l^{k+1} - \left(W_l^{k} + \omega^k\left(W_l^{k} - W_l^{k-1}\right)\right)\|(\text{Nesterov Acceleration})\\
&\quad\leqslant \rho M_{\mathbf{a}}\|z_l^{k+1} - z_l^{k}\| + \left(\rho M_{\mathbf{a}}^2 + \theta_l^{k+1}\right)\|W_l^{k+1} - W_l^{k}\|\\
&\quad+\left(\rho M_{\mathbf{a}}^2 + \theta_l^{k+1}\right)\|W_l^{k} - W_l^{k-1}\|(\text{Triangle Inequality and } \omega^k < 1).
\end{aligned} \tag{16}$$

On the other hand, the optimality condition of Eq. (3) yields

$$0 \in \partial\Omega_l\left(W_l^{k+1}\right) + \nabla_{\overline{W}_l^{k+1}}\phi(a_{l-1}^{k+1}, \overline{W}_l^{k+1}, z_l^{k}) + \theta_l^{k+1}\left(W_l^{k+1} - \overline{W}_l^{k+1}\right).$$

Therefore, there exists $g_{1,l}^{k+1} \in \partial_{W_l^{k+1}} F$ such that

$$\begin{aligned}
\|g_{1,l}^{k+1}\| \leqslant{}& \rho M_{\mathbf{a}}\|z_l^{k+1} - z_l^{k}\| + \left(\rho M_{\mathbf{a}}^2 + \theta_l^{k+1}\right)\|W_l^{k+1} - W_l^{k}\|\\
&+ \left(\rho M_{\mathbf{a}}^2 + \theta_l^{k+1}\right)\|W_l^{k} - W_l^{k-1}\|.
\end{aligned}$$

This shows that there exists $g_1^{k+1} = g_{1,1}^{k+1} \times g_{1,2}^{k+1} \times \cdots \times g_{1,L}^{k+1} \in \partial_{\mathbf{W}^{k+1}} F$ and $C_2 = \max\left(\rho M_{\mathbf{a}}, \rho M_{\mathbf{a}}^2 + \theta_1^{k+1}, \rho M_{\mathbf{a}}^2 + \theta_2^{k+1}, \cdots, \rho M_{\mathbf{a}}^2 + \theta_L^{k+1}\right)$ such that

$$\|g_l^{k+1}\| \leqslant C_2\left(\|\mathbf{W}^{k+1} - \mathbf{W}^{k}\| + \|\mathbf{z}^{k+1} - \mathbf{z}^{k}\| + \|\mathbf{W}^{k} - \mathbf{W}^{k-1}\|\right). \tag{17}$$

Otherwise, we have

$$\begin{aligned}
&\|\rho\left(W_l^{k+1} - \overline{W}_l^{k+1}\right)a_{l-1}^{k+1}(a_{l-1}^{k+1})^T - \rho(z_l^{k+1} - z_l^{k})(a_{l-1}^{k+1})^T\\
&-\theta_l^{k+1}\left(W_l^{k+1} - \overline{W}_l^{k+1}\right)\|\\
&\quad\leqslant \rho M_{\mathbf{a}}\|z_l^{k+1} - z_l^{k}\| + \left(\rho M_{\mathbf{a}}^2 + \theta_l^{k+1}\right)\|W_l^{k+1} - \overline{W}_l^{k+1}\|(\text{Eq.}(16))\\
&= \rho M_{\mathbf{a}}\|z_l^{k+1} - z_l^{k}\| + \left(\rho M_{\mathbf{a}}^2 + \theta_l^{k+1}\right)\|W_l^{k+1} - W_l^{k}\|\left(\overline{W}_l^{k+1} = W_l^{k}\right).
\end{aligned}$$

The optimality condition of Eq. (3) yields

$$0 \in \partial\Omega_l\left(W_l^{k+1}\right) + \nabla_{\overline{W}_l^{k+1}}\phi(a_{l-1}^{k+1}, \overline{W}_l^{k+1}, z_l^{k}) + \theta_l^{k+1}\left(W_l^{k+1} - \overline{W}_l^{k+1}\right).$$

By Eq. (15), we know that there exists $g_{1,l}^{k+1} \in \partial_{W_l^{k+1}} F$ such that

$$\|g_{1,l}^{k+1}\| \leqslant \rho M_{\mathbf{a}}\|z_l^{k+1} - z_l^{k}\| + \left(\rho M_{\mathbf{a}}^2 + \theta_l^{k+1}\right)\|W_l^{k+1} - W_l^{k}\|. \tag{18}$$

Combining Eq. (17) with Eq. (18), we show that there exists $g_1^{k+1} = g_{1,1}^{k+1} \times g_{1,2}^{k+1} \times \cdots \times g_{1,L}^{k+1} \in \partial_{\mathbf{W}^{k+1}} F$ and $C_2 = \max\left(\rho M_{\mathbf{a}}, \rho M_{\mathbf{a}}^2 + \theta_1^{k+1}, \rho M_{\mathbf{a}}^2 + \theta_2^{k+1}, \cdots, \rho M_{\mathbf{a}}^2 + \theta_L^{k+1}\right)$ such that

$$\|g_l^{k+1}\| \leqslant C_2\left(\|\mathbf{W}^{k+1} - \mathbf{W}^{k}\| + \|\mathbf{z}^{k+1} - \mathbf{z}^{k}\| + \|\mathbf{W}^{k} - \mathbf{W}^{k-1}\|\right).$$

Proof of Lemma 2

Proof. We add Eq. (7), Eq. (8), and Eq. (9) from $l = 1$ to L to obtain

$$\begin{aligned}
&F\left(\mathbf{W}^k, \mathbf{z}^k, \mathbf{a}^k\right) - F\left(\mathbf{W}^{k+1}, \mathbf{z}^{k+1}, \mathbf{a}^{k+1}\right)\\
&\geqslant \sum_{l=1}^{L}\left(\frac{\alpha_l^{k+1}}{2}\|W_l^{k+1} - W_l^{k}\|_2^2 + \frac{\gamma_l^{k+1}}{2}\|z_l^{k+1} - z_l^{k}\|_2^2\right) + \sum_{l=1}^{L-1}\frac{\delta_l^{k+1}}{2}\|a_l^{k+1} - a_l^{k}\|_2^2.
\end{aligned}$$

Let $C_5 = \min\left(\frac{\alpha_l^{k+1}}{2}, \frac{\gamma_l^{k+1}}{2}, \frac{\delta_l^{k+1}}{2}\right) > 0$, we have

$$\begin{aligned}
&F\left(\mathbf{W}^k, \mathbf{z}^k, \mathbf{a}^k\right) - F\left(\mathbf{W}^{k+1}, \mathbf{z}^{k+1}, \mathbf{a}^{k+1}\right)\\
&\geqslant C_5\left(\sum_{l=1}^{L}\left(\|W_l^{k+1} - W_l^{k}\|_2^2 + \|z_l^{k+1} - z_l^{k}\|_2^2\right) + \sum_{l=1}^{L-1}\|a_l^{k+1} - a_l^{k}\|_2^2\right)\\
&= C_5\left(\|\mathbf{W}^{k+1} - \mathbf{W}^{k}\|_2^2 + \|\mathbf{z}^{k+1} - \mathbf{z}^{k}\|_2^2 + \|\mathbf{a}^{k+1} - \mathbf{a}^{k}\|_2^2\right)\\
&\geqslant 0.
\end{aligned} \tag{19}$$

By Lemma 3 and a monotone sequence is convergent if it is bounded, then $F\left(\mathbf{W}^k, \mathbf{z}^k, \mathbf{a}^k\right)$ is convergent.

Proof of Theorem 1

Proof. By Lemma 3 (a), $\lim_{k\to\infty}\mathbf{W}^{k+1} - \mathbf{W}^k = 0$. By Lemma 3 (b), there exists a subsequence $\mathbf{W}^s$ such that $\mathbf{W}^s \to \mathbf{W}^*$, where W^* is a limit point. From Lemma 4, there exist $g_1^s \in \partial_{\mathbf{W}^s} F$ such that $\|g_1^s\| \to 0$ as $s \to \infty$. According to the definition of limiting subdifferential, we have $0 \in \partial_{\mathbf{W}^*} F$. In other words, $\mathbf{W}^*$ is a stationary point of F in Problem 2.

Proof of Theorem 2

Proof. In Algorithm 1, we prove this by the KL Property.

Firstly, we consider Eq. (4) and Eq. (6), by Lemma 3, $h_l\left(\overline{z}_l^{k+1} - \nabla\phi_{\overline{z}_l^{k+1}}/\rho\right) - a_l^k$ and $h_l(z^{k+1}) - \overline{a}_l^{k+1} + \nabla_{\overline{a}_l^{k+1}}\phi/\tau_l^{k+1}$ are bounded, i.e. there exist constants D_1 and D_2 such that

$$\begin{aligned}
&|h_l\left(\overline{z}_l^{k+1} - \nabla_{\overline{z}_l^{k+1}}\phi/\rho\right) - a_l^k| < D_1.\\
&\ |h_l(z^{k+1}) - \overline{a}_l^{k+1} + \nabla_{\overline{a}_l^{k+1}}\phi/\tau_l^{k+1}| < D_2.
\end{aligned}$$

Let $\varepsilon = \max(D_1, D_2)$, then the solutions to Eq. (4) and Eq. (6) are simplified as follows:

$$z_l^{k+1} \leftarrow \overline{z}_l^{k+1} - \nabla_{\overline{z}_l^{k+1}}\phi/\rho. \tag{20}$$

$$a_l^{k+1} \leftarrow \overline{a}_l^{k+1} - \nabla_{\overline{a}_l^{k+1}}\phi/\tau_l^{k+1}. \tag{21}$$

This is because $h_l(z_l^{k+1}) - \varepsilon \leqslant a_l^k \leqslant h_l(z_l^{k+1}) + \varepsilon$ and $h_l(z_l^{k+1}) - \varepsilon \leqslant a_l^{k+1} \leqslant h_l(z_l^{k+1}) + \varepsilon$ hold in Eq. (4) and Eq. (6), respectively.

Next, we prove that given $\varepsilon = \max(D_1, D_2)$, there exists $C_3 = \max\left(\rho M_{\mathbf{W}}^2 + \tau_1^{k+1}, \rho M_{\mathbf{W}}^2 + \tau_2^{k+1}, \rho M_{\mathbf{W}}^2 + \tau_3^{k+1}, \cdots, \rho M_{\mathbf{W}}^2 + \tau_{L-1}^{k+1},\right.$ $\left. 2\rho M_{\mathbf{W}} M_{\mathbf{a}} + \rho M_{\mathbf{z}}\right)$, some $g_3^{k+1} \in \partial_{\mathbf{z}^{k+1}} F$ and $g_4^{k+1} \in \partial_{\mathbf{a}^{k+1}} F$ such that

$$\|g_3^{k+1}\| = 0,$$
$$\|g_4^{k+1}\| \leqslant C_3\left(\|\mathbf{a}^{k+1} - \mathbf{a}^k\| + \|\mathbf{a}^k - \mathbf{a}^{k-1}\| + \|\mathbf{W}^{k+1} - \mathbf{W}^k\| + \|\mathbf{z}^{k+1} - \mathbf{z}^k\|\right).$$

As shown in [33,34],

$$\partial_{\mathbf{z}^{k+1}} F = \partial_{z_1^{k+1}} F \times \partial_{z_2^{k+1}} F \times \cdots \times \partial_{z_L^{k+1}} F,$$
$$\nabla_{\mathbf{a}^{k+1}} F = \nabla_{a_1^{k+1}} F \times \nabla_{a_2^{k+1}} F \times \cdots \times \nabla_{a_{L-1}^{k+1}} F,$$

where $\times$ denotes Cartesian Product.

For $z_l^{k+1} (l < L)$, according to Line 18 of Algorithm 1, no matter. $F\left(\mathbf{W}_{\leqslant l}^{k+1}, \mathbf{z}_{\leqslant l}^{k+1}, \mathbf{a}_{\leqslant l-1}^{k+1}\right) \geqslant F\left(\mathbf{W}_{\leqslant l}^{k+1}, \mathbf{z}_{\leqslant l-1}^{k+1}, \mathbf{a}_{\leqslant l-1}^{k+1}\right)$ or not, we have

$$\begin{aligned}\partial_{z_l^{k+1}} F &= \nabla_{z_l^{k+1}} \phi\left(a_{l-1}^{k+1}, W_l^{k+1}, z_l^{k+1}\right)\\ &= \nabla_{z_l^{k+1}} \phi\left(a_{l-1}^{k+1}, W_l^{k+1}, z_l^{k+1}\right) - \nabla_{\bar{z}_l^{k+1}} \phi\left(a_{l-1}^{k+1}, W_l^{k+1}, \bar{z}_l^{k+1}\right)\\ &\quad -\rho\left(z_l^{k+1} - \bar{z}_l^{k+1}\right)(\text{Eq.}(20)) = 0.\end{aligned}$$

For z_L^{k+1}, according to Line 12 of Algorithm 1, no matter $F\left(\mathbf{W}_{\leqslant L}^{k+1}, \mathbf{z}_{\leqslant L}^{k+1}, \mathbf{a}_{\leqslant L-1}^{k+1}\right) \geqslant F\left(\mathbf{W}_{\leqslant L}^{k+1}, \mathbf{z}_{\leqslant L-1}^{k+1}, \mathbf{a}_{\leqslant L-1}^{k+1}\right)$ or not, we have

$$\begin{aligned}\partial_{z_L^{k+1}} F &= \nabla_{z_L^{k+1}} \phi\left(a_{L-1}^{k+1}, W_L^{k+1}, z_L^{k+1}\right) + \partial R(z_L^{k+1}; y)\\ &= \nabla_{z_L^{k+1}} \phi\left(a_{L-1}^{k+1}, W_L^{k+1}, z_L^{k+1}\right) + \partial R(z_L^{k+1}; y)\\ &\quad + \nabla_{\bar{z}_L^{k+1}} \phi\left(a_{L-1}^{k+1}, W_L^{k+1}, \bar{z}_L^{k+1}\right)\\ &\quad + \rho\left(z_L - \bar{z}_L^{k+1}\right) - \nabla_{\bar{z}_L^{k+1}} \phi\left(a_{L-1}^{k+1}, W_L^{k+1}, \bar{z}_L^{k+1}\right) - \rho\left(z_L^{k+1} - \bar{z}_L^{k+1}\right)\\ &= \nabla_{z_L^{k+1}} \phi\left(a_{L-1}^{k+1}, W_L^{k+1}, z_L^{k+1}\right) - \nabla_{\bar{z}_L^{k+1}} \phi\left(a_{L-1}^{k+1}, W_L^{k+1}, \bar{z}_L^{k+1}\right)\\ &\quad - \rho\left(z_L^{k+1} - \bar{z}_L^{k+1}\right)\left(0 \in \partial R(\bar{z}_L^{k+1}; y) + \nabla_{\bar{z}_L^{k+1}} \phi\left(a_{L-1}^{k+1}, W_L^{k+1}, \bar{z}_L^{k+1}\right)\right.\\ &\quad \left.+\rho\left(z_L^{k+1} - \bar{z}_L^{k+1}\right) \text{by the optimality condition of Eq.}(5)\right) = 0.\end{aligned}$$

Therefore, there exists $g_{3,l}^{k+1} = \nabla_{z_l^{k+1}} F$ such that $\|g_{3,l}^{k+1}\| = 0$. This shows that there exists $g_3^{k+1} = g_{3,1}^{k+1} \times g_{3,2}^{k+1} \times \cdots \times g_{3,L}^{k+1} = \nabla_{\mathbf{z}^{k+1}} F$ such that

$$\|g_3^{k+1}\| = 0. \tag{22}$$

For a_l^{k+1}, we have

$$\begin{aligned}&\partial_{a_l^{k+1}} F = \nabla_{a_l^{k+1}} \phi\left(a_l^{k+1}, W_{l+1}^k, z_{l+1}^{k+1}\right)\\ &= \nabla_{a_l^{k+1}} \phi\left(a_l^{k+1}, W_{l+1}^{k+1}, z_{l+1}^{k+1}\right) - \nabla_{\bar{a}_l^{k+1}} \phi\left(\bar{a}_l^{k+1}, W_{l+1}^k, z_{l+1}^k\right)\\ &-\tau_l^{k+1}(a_l^{k+1} - \bar{a}_l^{k+1})(\text{Eq.}(21))\\ &= \rho\left(W_{l+1}^{k+1}\right)^T\left(W_{l+1}^{k+1} a_l^{k+1} - z_{l+1}^{k+1}\right) - \rho\left(W_{l+1}^k\right)^T\left(W_{l+1}^k \bar{a}_l^{k+1} - z_{l+1}^k\right)\\ &-\tau_l^{k+1}(a_l^{k+1} - \bar{a}_l^{k+1}) = \rho\left(W_{l+1}^{k+1}\right)^T W_{l+1}^{k+1}(a_l^{k+1} - \bar{a}_l^{k+1}) + \rho\left(W_{l+1}^{k+1}\right)^T\\ &\left(W_{l+1}^{k+1} - W_{l+1}^k\right)\bar{a}_l^{k+1} + \rho\left(W_{l+1}^{k+1} - W_{l+1}^k\right)^T W_{l+1}^k \bar{a}_l^{k+1} - \rho\left(W_{l+1}^{k+1}\right)^T\\ &(z_{l+1}^{k+1} - z_{l+1}^k) - \rho\left(W_{l+1}^{k+1} - W_{l+1}^k\right)^T z_{l+1}^k - \tau_l^{k+1}(a_l^{k+1} - \bar{a}_l^{k+1}).\end{aligned}$$

Therefore

$$\begin{aligned}&\|\partial_{a_l^{k+1}} F\| \leqslant \rho\|W_{l+1}^{k+1}\|\|W_{l+1}^{k+1}\|\|a_l^{k+1} - \bar{a}_l^{k+1}\| + \rho\|W_{l+1}^{k+1}\|\|W_{l+1}^{k+1}\\ &-W_{l+1}^k\|\|\bar{a}_l^{k+1}\| + \rho\|W_{l+1}^{k+1} - W_{l+1}^k\|\|W_{l+1}^k\|\|\bar{a}_l^{k+1}\| + \rho\|W_{l+1}^{k+1}\|\|z_{l+1}^{k+1}\\ &-z_{l+1}^k\| + \rho\|W_{l+1}^{k+1} - W_{l+1}^k\|\|z_{l+1}^k\| + \tau_l^{k+1}\|a_l^{k+1} - \bar{a}_l^{k+1}\|\\ &(\text{Triangle Inequality and Cauthy} - \text{Schwarz Inequality})\\ &\leqslant \rho M_{\mathbf{W}}^2\|a_l^{k+1} - \bar{a}_l^{k+1}\| + \rho M_{\mathbf{W}}\|W_{l+1}^{k+1} - W_{l+1}^k\|M_{\mathbf{a}} + \rho\|W_{l+1}^{k+1}\\ &-W_{l+1}^k\|M_{\mathbf{W}} M_{\mathbf{a}} + \rho M_{\mathbf{W}}\|z_{l+1}^{k+1} - z_{l+1}^k\| + \rho\|W_{l+1}^{k+1} - W_{l+1}^k\|M_{\mathbf{z}}\\ &+\tau_l^{k+1}\|a_l^{k+1} - \bar{a}_l^{k+1}\| \ (\text{Lemma 3}) = \left(\rho M_{\mathbf{W}}^2 + \tau_l^{k+1}\right)\|a_l^{k+1} - \bar{a}_l^{k+1}\|\\ &\quad +(2\rho M_{\mathbf{W}} M_{\mathbf{a}} + \rho M_{\mathbf{z}})\|W_{l+1}^{k+1} - W_{l+1}^k\| + \rho M_{\mathbf{W}}\|z_{l+1}^{k+1} - z_{l+1}^k\|.\end{aligned}$$

According to Line 22 of Algorithm 1, if $F\left(\mathbf{W}_{\leqslant l}^{k+1}, \mathbf{z}_{\leqslant l}^{k+1}, \mathbf{a}_{\leqslant l}^{k+1}\right) < F\left(\mathbf{W}_{\leqslant l}^{k+1}, \mathbf{z}_{\leqslant l}^{k+1}, \mathbf{a}_{\leqslant l-1}^{k+1}\right)$, then we have

$$\begin{aligned}&\|\partial_{a_l^{k+1}} F\| \leqslant \left(\rho M_{\mathbf{W}}^2 + \tau_l^{k+1}\right)\|a_l^{k+1} - a_l^k - (a_l^k - a_l^{k-1})\omega^k\|\\ &+(2\rho M_{\mathbf{W}} M_{\mathbf{a}} + \rho M_{\mathbf{z}})\|W_{l+1}^{k+1} - W_{l+1}^k\| + \rho M_{\mathbf{W}}\|z_{l+1}^{k+1} - z_{l+1}^k\|\\ &\quad (\text{Nestrov Acceleration}); \leqslant \left(\rho M_{\mathbf{W}}^2 + \tau_l^{k+1}\right)\|a_l^{k+1} - a_l^k\|\\ &+\left(\rho M_{\mathbf{W}}^2 + \tau_l^{k+1}\right)\|a_l^k - a_l^{k-1}\| + (2\rho M_{\mathbf{W}} M_{\mathbf{a}} + \rho M_{\mathbf{z}})\|W_{l+1}^{k+1} - W_{l+1}^k\|\\ &+\rho M_{\mathbf{W}}\|z_{l+1}^{k+1} - z_{l+1}^k\|(\text{Triangle Inequality and } \omega^k < 1).\end{aligned}$$

Therefore, there exists $g_{4,l}^{k+1} \in \partial_{a_l^{k+1}} F$ such that

$$\begin{aligned}&\|g_{4,l}^{k+1}\| \leqslant \left(\rho M_{\mathbf{W}}^2 + \tau_l^{k+1}\right)\|a_l^{k+1} - a_l^k\| + \left(\rho M_{\mathbf{W}}^2 + \tau_l^{k+1}\right)\|a_l^k - a_l^{k-1}\|\\ &+(2\rho M_{\mathbf{W}} M_{\mathbf{a}} + \rho M_{\mathbf{z}})\|W_{l+1}^{k+1} - W_{l+1}^k\| + \rho M_{\mathbf{W}}\|z_{l+1}^{k+1} - z_{l+1}^k\|.\end{aligned} \tag{23}$$

Otherwise,

$$\begin{aligned}&\|\partial_{a_l^{k+1}} F\| \leqslant \left(\rho M_{\mathbf{W}}^2 + \tau_l^{k+1}\right)\|a_l^{k+1} - a_l^k\| + (2\rho M_{\mathbf{W}} M_{\mathbf{a}} + \rho M_{\mathbf{z}})\|W_{l+1}^{k+1}\\ &\quad - W_{l+1}^k\| + \rho M_{\mathbf{W}}\|z_{l+1}^{k+1} - z_{l+1}^k\| \ (\bar{a}_l^{k+1} = a_l^k).\end{aligned}$$

Therefore, there exists $g_{4,l}^{k+1} \in \partial_{a_l^{k+1}} F$ such that

$$\begin{aligned}&\|g_{4,l}^{k+1}\| \leqslant \left(\rho M_{\mathbf{W}}^2 + \tau_l^{k+1}\right)\|a_l^{k+1} - a_l^k\|\\ &\quad + (2\rho M_{\mathbf{W}} M_{\mathbf{a}} + \rho M_{\mathbf{z}})\|W_{l+1}^{k+1} - W_{l+1}^k\| + \rho M_{\mathbf{W}}\|z_{l+1}^{k+1}\\ &\quad - z_{l+1}^k\|.\end{aligned} \tag{24}$$

Combining Eq. (23) and Eq. (24), we show that there exists $g_4^{k+1} = g_{4,1}^{k+1} \times g_{4,2}^{k+1} \times \cdots \times g_{4,L}^{k+1} \in \partial_{\mathbf{a}^{k+1}} F$ and $C_3 = \max\left(\rho M_{\mathbf{W}}^2 + \tau_1^{k+1},\right.$ $\rho M_{\mathbf{W}}^2 + \tau_2^{k+1}, \rho M_{\mathbf{W}}^2 + \tau_3^{k+1}, \cdots, \rho M_{\mathbf{W}}^2 + \left.\tau_{L-1}^{k+1}, 2\rho M_{\mathbf{W}} M_{\mathbf{a}} + \rho M_{\mathbf{z}}\right)$ such that

$$\|g_4^{k+1}\| \leqslant C_3\left(\|\mathbf{a}^{k+1} - \mathbf{a}^k\| + \|\mathbf{a}^k - \mathbf{a}^{k-1}\| + \|\mathbf{W}^{k+1} - \mathbf{W}^k\| + \|\mathbf{z}^{k+1} - \mathbf{z}^k\|\right). \tag{25}$$

Combining Lemma 4, Eq. (22) and Eq. (25), we prove that there exists $g^{k+1} \in \partial F\left(\mathbf{W}^{k+1}, \mathbf{z}^{k+1}, \mathbf{a}^{k+1}\right) = \{\partial_{\mathbf{W}^{k+1}} F, \partial_{\mathbf{z}^{k+1}} F, \partial_{\mathbf{a}^{k+1}} F\}$ and $C_4 = \max(C_2, C_3, \rho)$ such that

$$\begin{aligned}\|g^{k+1}\| \leqslant C_4&\left(\|\mathbf{a}^{k+1} - \mathbf{a}^k\| + \|\mathbf{a}^k - \mathbf{a}^{k-1}\| + \|\mathbf{W}^{k+1} - \mathbf{W}^k\|\right.\\ &\left.+ \|\mathbf{W}^k - \mathbf{W}^{k-1}\| + \|\mathbf{z}^{k+1} - \mathbf{z}^k\|\right).\end{aligned} \tag{26}$$

Finally, we prove the linear convergence rate by the KL Property given Eq. (26) and Eq. (19). Because F is locally strongly convex with a constant μ, F satisfies the KL Property by Lemma 6. Let $F^* = F(\mathbf{W}^*, \mathbf{z}^*, \mathbf{a}^*)$ be the convergent value of F, by Lemma 2,

$F\left(\mathbf{W}^k, \mathbf{z}^k, \mathbf{a}^k\right) \to F^*$, then for any $\eta_1 > 0$ there exists $k_2 \in \mathbb{N}$ such that it holds for $k > k_2$ that $F^* < F\left(\mathbf{W}^k, \mathbf{z}^k, \mathbf{a}^k\right) < F^* + \eta_1$. Also by Lemma 3a) and Eq. (26), $g^{k+1} \to 0$ as $k \to \infty$, then for any $\eta_2 > 0$ there exists $k_3 \in \mathbb{N}$, such that it holds for $k > k_3$ that $\|g^{k+1}\| < \eta_2$. Therefore, for any $k > k_1 = \max(k_2, k_3)$, $\left(\mathbf{W}^k, \mathbf{z}^k, \mathbf{a}^k\right) \in \{(\mathbf{W}, \mathbf{z}, \mathbf{a}) : |F^* < F(\mathbf{W}, \mathbf{z}, \mathbf{a}) < F^* + \eta_1 \cap \exists g \in F(\mathbf{W}, \mathbf{z}, \mathbf{a})\ s.t.\ \|g\| < \eta_2\}$. By the KL Property and Lemma 6, it holds that

$$
\begin{aligned}
1 &\leqslant \|g^{k+1}\| / \left(\mu\sqrt{F\left(\mathbf{W}^{k+1}, \mathbf{z}^{k+1}, \mathbf{a}^{k+1}\right) - F^*}\right) \\
&\leqslant C_4\Big(\|\mathbf{a}^{k+1} - \mathbf{a}^k\| + \|\mathbf{a}^k - \mathbf{a}^{k-1}\| + \|\mathbf{W}^{k+1} - \mathbf{W}^k\| + \|\mathbf{W}^k - \mathbf{W}^{k-1}\| \\
&\quad + \|\mathbf{z}^{k+1} - \mathbf{z}^k\|\Big) / \left(\mu\sqrt{F\left(\mathbf{W}^{k+1}, \mathbf{z}^{k+1}, \mathbf{a}^{k+1}\right) - F^*}\right) (\text{Eq.}(26)) \\
&\leqslant C_4^2\Big(\|\mathbf{a}^{k+1} - \mathbf{a}^k\| + \|\mathbf{a}^k - \mathbf{a}^{k-1}\| + \|\mathbf{W}^{k+1} - \mathbf{W}^k\| \\
&\quad + \|\mathbf{W}^k - \mathbf{W}^{k-1}\| + \|\mathbf{z}^{k+1} - \mathbf{z}^k\|\Big)^2 / \left(\mu^2\left(F\left(\mathbf{W}^{k+1}, \mathbf{z}^{k+1}, \mathbf{a}^{k+1}\right) - F^*\right)\right) \\
&\leqslant \Big(5C_4^2\Big(\|\mathbf{a}^{k+1} - \mathbf{a}^k\|_2^2 + \|\mathbf{a}^k - \mathbf{a}^{k-1}\|_2^2 + \|\mathbf{W}^{k+1} - \mathbf{W}^k\|_2^2 \\
&\quad + \|\mathbf{W}^k - \mathbf{W}^{k-1}\|_2^2 + \|\mathbf{z}^{k+1} - \mathbf{z}^k\|_2^2\Big)\Big) / \left(\mu^2\left(F\left(\mathbf{W}^{k+1}, \mathbf{z}^{k+1}, \mathbf{a}^{k+1}\right) - F^*\right)\right) \\
&\quad (\text{Mean Inequality}) \\
&\leqslant \Big(5C_4^2\Big(F\left(\mathbf{W}^{k-1}, \mathbf{z}^{k-1}, \mathbf{a}^{k-1}\right) \\
&\quad - F\left(\mathbf{W}^{k+1}, \mathbf{z}^{k+1}, \mathbf{a}^{k+1}\right)\Big)\Big) / \left(C_5\mu^2\left(F\left(\mathbf{W}^{k+1}, \mathbf{z}^{k+1}, \mathbf{a}^{k+1}\right) - F^*\right)\right) (\text{Eq.}(19)).
\end{aligned}
$$

This indicates that

$$
\begin{aligned}
&\left(C_5\mu^2 + 5C_4^2\right)\left(F\left(\mathbf{W}^{k+1}, \mathbf{z}^{k+1}, \mathbf{a}^{k+1}\right) - F^*\right) \\
&\quad \leqslant 5C_4^2\left(F\left(\mathbf{W}^{k-1}, \mathbf{z}^{k-1}, \mathbf{a}^{k-1}\right) - F^*\right).
\end{aligned}
$$

Let $0 < C_1 = \frac{5C_4^2}{C_5\mu^2 + 5C_4^2} < 1$, we have

$$
F\left(\mathbf{W}^{k+1}, \mathbf{z}^{k+1}, \mathbf{a}^{k+1}\right) - F^* \leqslant C_1\left(F\left(\mathbf{W}^{k-1}, \mathbf{z}^{k-1}, \mathbf{a}^{k-1}\right) - F^*\right).
$$

So in summary, for any ρ, there exist $\varepsilon = \max(D_1, D_2), k_1 = \max(k_2, k_3)$, and $0 < C_1 = \frac{5C_4^2}{C_5\mu^2 + 5C_4^2} < 1$ such that

$$
F\left(\mathbf{W}^{k+1}, \mathbf{z}^{k+1}, \mathbf{a}^{k+1}\right) - F^* \leqslant C_1\left(F\left(\mathbf{W}^{k-1}, \mathbf{z}^{k-1}, \mathbf{a}^{k-1}\right) - F^*\right).
$$

for $k > k_1$. In other words, the linear convergence rate is proven.

References

[1] Armin Askari, Geoffrey Negiar, Rajiv Sambharya, and Laurent El Ghaoui. Lifted neural networks. NIPS Workshop on Optimization, 2017.
[2] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. In SIAM journal on imaging sciences, volume 2, pages 183–202. SIAM, 2009.
[3] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In Yves Lechevallier and Gilbert Saporta, editors, Proceedings of COMPSTAT'2010, pages 177–186. Physica-Verlag HD, 2010.
[4] Miguel Carreira-Perpinan and Weiran Wang. Distributed optimization of deeply nested systems. In Artificial Intelligence and Statistics, pages 10–19, 2014.
[5] Lei Chen, Zhengdao Chen, Joan Bruna, On graph neural networks versus graph-augmented mlps, in: Ninth International Conference on Learning Representations, 2021.
[6] Anna Choromanska, Benjamin Cowen, Sadhana Kumaravel, Ronny Luss, Mattia Rigotti, Irina Rish, Paolo Diachille, Viatcheslav Gurev, Brian Kingsbury, Ravi Tejwani, et al. Beyond backprop: Online alternating minimization with auxiliary variables. In International Conference on Machine Learning, pages 1193–1202. PMLR, 2019.
[7] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In Advances in neural information processing systems, pages 3123–3131, 2015.
[8] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 12(Jul):2121–2159, 2011.
[9] Saeed Ghadimi, Guanghui Lan, Accelerated gradient methods for nonconvex nonlinear and stochastic programming, Math. Programm. 156 (1–2) (2016) 59–99.
[10] Xavier Glorot, Yoshua Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the thirteenth international conference on artificial intelligence and statistics, 2010, pp. 249–256.
[11] Gauri Jagatap and Chinmay Hegde. Learning relu networks via alternating minimization. arXiv preprint arXiv:1806.07863, 2018.
[12] Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In International Conference on Learning Representations (ICLR) Poster, 2015.
[13] Tim Tsz-Kit Lau, Jinshan Zeng, Baoyuan Wu, and Yuan Yao. A proximal block coordinate descent algorithm for deep neural network training. International Conference on Learning Representations Workshop, 2018.
[14] Hongyi Li, Junxiang Wang, Yongchao Wang, Yue Cheng, Liang Zhao, Community-based layerwise distributed training of graph convolutional networks, in: NeurIPS 2021 Workshop on Optimization for Machine Learning (OPT 2021), 2021.
[15] Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. In International Conference on Learning Representations, 2018.
[16] Andrew L. Maas, Awni Y. Hannun, Andrew Y. Ng, Rectifier nonlinearities improve neural network acoustic models. In ICML Workshop on Deep Learning for Audio, Speech and Language Processing. Citeseer (2013).
[17] Boris T Polyak, Some methods of speeding up the convergence of iteration methods, USSR Computational Mathematics and Mathematical Physics 4 (5) (1964) 1–17.
[18] Linbo Qiao, Tao Sun, Hengyue Pan, Dongsheng Li, Inertial proximal deep learning alternating minimization for efficient neutral network training, in: ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) IEEE, 2021, pp. 3895–3899.
[19] Sashank J. Reddi, Satyen Kale, Sanjiv Kumar, On the convergence of adam and beyond, in: International Conference on Learning Representations, 2018.
[20] Herbert Robbins, S. Monro, A stochastic approximation method, annals math, Statistics 22 (1951) 400–407.
[21] R. Tyrrell Rockafellar, Roger J-B Wets, Variational analysis, vol. 317, Springer Science & Business Media, 2009.
[22] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. In Nature, volume 323, page 533. Nature Publishing Group, 1986.
[23] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, Tina Eliassi-Rad, Collective classification in network data, AI Magazine 29 (2008) 93–106.
[24] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. Relational Representation Learning Workshop (R2L 2018), NeurIPS, 2018.
[25] Ilya Sutskever, James Martens, George Dahl, Geoffrey Hinton, On the importance of initialization and momentum in deep learning, in: International conference on machine learning, 2013, pp. 1139–1147.
[26] Yu Tang, Zhigang Kan, Dequan Sun, Linbo Qiao, Jingjing Xiao, Zhiquan Lai, and Dongsheng Li. Admmirnn: Training rnn with stable convergence via an efficient admm approach. In Frank Hutter, Kristian Kersting, Jefrey Lijffijt, and Isabel Valera, editors, Machine Learning and Knowledge Discovery in Databases, pages 3–18, Cham, 2021. Springer International Publishing.
[27] Gavin Taylor, Ryan Burmeister, Zheng Xu, Bharat Singh, Ankit Patel, and Tom Goldstein. Training neural networks without gradients: A scalable admm approach. In International Conference on Machine Learning, pages 2722–2731, 2016.
[28] T. Tieleman, G. Hinton, Divide the gradient by a running average of its recent magnitude coursera: Neural networks for machine learning, in: Technical report, University of Toronto, 2017.
[29] Junxiang Wang, Zheng Chai, Yue Cheng, Liang Zhao, Toward model parallelism for deep neural network based on gradient-free admm framework, in: 2020 IEEE International Conference on Data Mining (ICDM) IEEE, 2020, pp. 591–600.
[30] Junxiang Wang, Hongyi Li, Zheng Chai, Yongchao Wang, Yue Cheng, and Liang Zhao. Towards quantized model parallelism for graph-augmented mlps based on gradient-free admm framework, 2021.
[31] Junxiang Wang, Yu. Fuxun, Xiang Chen, Liang Zhao, Admm for efficient deep learning with global convergence, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019, pp. 111–119.
[32] Junxiang Wang and Liang Zhao. Nonconvex generalization of alternating direction method of multipliers for nonlinear equality constrained problems. Results in Control and Optimization, page 100009, 2021.
[33] Yu. Wang, Wotao Yin, Jinshan Zeng, Global convergence of admm in nonconvex nonsmooth optimization, J. Sci. Comput. (2015) 1–35.
[34] Xu. Yangyang, Wotao Yin, A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion, SIAM J. Imaging Sci. 6 (3) (2013) 1758–1789.

[35] Babak Zamanlooy, Mitra Mirhassani, Efficient vlsi implementation of neural networks with hyperbolic tangent activation function, IEEE Transactions on Very Large Scale Integration (VLSI) Systems 22 (1) (2014) 39–48.
[36] Matthew D Zeiler. Adadelta: an adaptive learning rate method. preprint, 2012.
[37] Jinshan Zeng, Tim Tsz-Kit Lau, Shaobo Lin, Yuan Yao, Global convergence of block coordinate descent in deep learning, in: Proceedings of the 36th International Conference on Machine Learning PMLR, 2019, pp. 7313–7323.
[38] G. Zhang and W.B. Kleijn. Training deep neural networks via optimization over graphs. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4119–4123, April 2018.
[39] Ziming Zhang and Matthew Brand. Convergent block coordinate descent for training tikhonov regularized deep neural networks. In Advances in Neural Information Processing Systems, pages 1721–1730, 2017.
[40] Ziming Zhang, Yuting Chen, Venkatesh Saligrama, Efficient training of very deep neural networks for supervised hashing, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1487–1495.
[41] Juntang Zhuang, Tommy Tang, Yifan Ding, Sekhar C Tatikonda, Nicha Dvornek, Xenophon Papademetris, and James Duncan. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. Advances in Neural Information Processing Systems, 33, 2020.

Junxiang Wang received the B.S degree and the Master degree from East China Normal University, Shanghai, China in 2012, and George Mason University, Virginia, United States in 2020, respectively. He is a Ph.D. candidate in the Department of Computer Science at Emory University supervised by Professor Liang Zhao. His research focuses on data mining in social media and nonconvex optimization in deep learning.

Hongyi Li received the B.E. degree in Telecommunication Engineering from Xidian University, Xi'an, China, in 2019. She is a Ph.D. student at the State Key Laboratory of ISN in Xidian University, China. Her research interests include graph learning, optimization algorithms, and their applications in wireless communication systems.

Liang Zhao is an assistant professor at the Department of Computer Science at Emory University. He obtained his Ph.D. degree in 2016 from Computer Science Department at Virginia Tech in the United States. His research interests include data mining, artificial intelligence, and machine learning, with special interests in spatiotemporal and network data mining, deep learning on graphs, nonconvex optimization, and interpretable machine learning.