The Global Convergence of the Alternating Minimization Algorithm for Deep Neural Network Problems*

Junxiang Wang, Fuxun Yu, Xiang Chen& Liang Zhao
George Mason University
Fairfax, VA 22030, USA
{jwang40,fyu2,xchen26,lzhao9}@gmu.edu

Abstract

In recent years, stochastic gradient descent(SGD) is a dominant optimization method for training deep neural networks. But the SGD suffers from several limitations including lack of theoretical guarantees, gradient vanishing, poor conditioning and the non-differentiability of activation functions, which motives the development of alternating minimization methods. However, there are still two challenges needed to overcome: difficulty in obtaining global minimum in subproblems, and expensive cost of matrix inversion between layers. In this paper, we propose a novel deep learning alternating minimization(DLAM) algorithm to deal with those two challenges. Furthermore, detail proofs are provided to validate the global convergence of our DLAM algorithm under mild conditions. Experiments on real-world datasets demonstrate the effectiveness of our DLAM algorithm.

1 Introduction

During the past few decades, stochastic gradient descent(SGD) is a major optimization method for training deep neural networks. The mechanism of the SGD is to split a dataset into multiple batches and then optimizes them sequentially by gradient descent in each epoch. The popularity of SGD consists in two advantages: firstly, it is simple to implement; Secondly, the SGD can be applied in online settings where new coming training data are used to train models. However, while many papers provided solid theoretical guarantees on the convergence of SGD[10, 14, 18], the assumptions of the proofs cannot be applied in deep neural networks, problems of which are highly nonlinear and nonconvex. Aside from lack of theoretical guarantees, several additional drawbacks restrict

*Preprint: work in progress.

the applications of SGD further including gradient vanishing, poor conditioning and the non-differentiability of activation functions[19].

To tackle the above-mentioned intrinsic drawbacks of gradient descent optimization methods, recently, alternating minimization methods have started to attract the attention of researchers to solve deep learning problems, where the loss function of a deep neural network is reformulated as a nested function associated with multiple linear and nonlinear transformations across multi-layers. A typical way to represent such a neural network is to decompose its nested structure into a series of linear and non-linear equality constraints by introducing auxiliary variables. These linear and non-linear equality constraints generate multiple subproblems, which can be minimized alternately. Recent works have proposed several alternating minimization methods including Alternating Direction Method of Multipliers(ADMM)[19] and block coordinate descent(BCD)[9]. Empirical evaluations demonstrated high accuracy on the test sets and good scalability compared with SGD thanks to parallelism. Moreover, these methods avoid gradient vanishing problems and allow for non-differentiable activation functions such as binarized neural networks[6].

However, as an emerging domain, alternating minimization methods still suffer from several challenges including: **1. No global convergence with mild conditions.** Among all of existing alternating minimization methods for deep learning, only few have provided convergence analysis yet under strict conditions. The reason is that many subproblems generated by nonlinear-equality constraints are highly nonconvex and difficult to obtain global minimum. For example, it is impossible to minimize output update exactly in ADMM with nonlinear smooth sigmoid functions, which requires lookup tables[19]. **2. Huge computational cost with high dimensions.** Existing alternating minimization methods typically requires matrix inversion due to the computation of auxiliary variables. For example, the weight update and the activation update need to compute the pseudoinverse of a matrix in ADMM[19]. However, the operation of matrix inversion needs many subiterations in order to get a precise inverse matrix, and hence is very time-consuming. As the number of neurons and training samples grow, the computational cost of the matrix inversion increases quadratically.

In order to simultaneously address these technical problems, we propose a novel deep learning alternating minimization(DLAM) algorithm, which is flexible to use in different network structures. Specifically, we give a new formulation of the deep neural network problem where the nonlinear activation functions are replaced equivalently by inequality constraints. In this formulation, all subproblems contain global minimum and can be solved efficiently. In order to handle expensive time cost of the matrix inversion, we avoid the operation of matrix inversion by the quadratic approximation technique and the backtracking algorithm, which speed up the convergence of the DLAM algorithm. Moreover, while some works require strict and complex conditions such as Kurdyka-ojasiewicz (KL) properties[11] to prove convergence, we present simple and mild conditions to guarantee convergence, which covers most common loss functions and activation functions. Last but not least, the choice of parameters has no effect on the convergence of our DLAM algorithm. Our contribution in this paper includes:

- We propose a novel formulation of the fully-connected deep neural network problem. The nonlinear activation functions are replaced by equivalent inequality constraints. This formulation ensures that all subproblems have global minimums.
- We present a novel DLAM algorithm and discuss every subproblem in detail. A quadratic approximation technique and a backtracking algorithm are utilized to avoid the matrix inversion. Every subproblem has a closed-form solution and hence the DLAM is efficient.
- We investigate the convergence properties of the DLAM algorithm. The model assumptions are highly mild such that most deep learning frameworks fit into our assumptions with no difficulty. The DLAM algorithm is guaranteed to converge to a stationary solution of the original problem.
- We conduct experiments on real-world datasets to validate our DLAM algorithm. Experiments on two benchmark datasets show that our DLAM outperform the ADMM by a large margin.

The rest of the paper is organized as follows. In Section 2, we summarize recent research work related to this paper. In Section 3.2, we present the problem formulation and DLAM algorithm. In Section 4, we introduce the main convergence results of the DLAM algorithm. In Section 5, extensive experiments are conducted to validate the convergence and effectiveness of our DLAM. Section 6 concludes by summarizing the whole paper.

2 Related Work

All existing works related to optimization methods in deep neural network problems are categorized as two major classes: stochastic gradient descent methods and alternating minimization methods. This research is surveyed in the following.

Stochastic gradient descent methods: The renaissance of SGD can be traced back to 1951, as proposed by Robbins and Monro[15]. Then the famous back-propagation algorithm was introduced by Rumelhart et al.[17]. Since then, many variants of SGD methods have been presented: for example, Polyak proposed the Polyak momentum to accelerate the convergence of iterative methods[13]. Sutskever et al. highlighted the importance of Nesterov momentum and initialization[18]. Many well-known SGD with adaptive learning rates were proposed and were prevalent in the deep learning community such as AdaGrad[8], RMSProp[21], Adam[10] and AMSGrad[14].

Alternating minimization methods for deep learning: Previous works on the application of alternating minimization algorithm in the deep learning problems can be categorized into two types. Some papers proposed alternating minimization algorithms on specific applications. For example, Taylor et al. presented an Alternating Direction Method of Multipliers(ADMM) to transform

Table 1: Important Notations and descriptions

Notations	Descriptions
L	Number of layers.
W_l	The weight vector in the $l-th$ layer.
b_l	The intercept in the $l-th$ layer.
z_l	The temporary variable 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 risk function in the $L-th$ layer.
$\Omega_l(W_l)$	The regularization term in the l-th layer.
ε_l	The tolerance of the nonlinear mapping in the l-th layer.

a fully-connected neural network problem into an equality-constrained problem, where many subproblems split by ADMM can be solved in parallel[19]. Zhang et al. handled very deep supervised hashing(VDSH) problem by the ADMM algorithm to overcome issues of vanishing gradients and computational efficiency[25]. Zhang and W. Bastiaan trained a deep neural network by ADMM over a graph[23].Askari et al. introduced a new framework of multilayer feedforward neural networks and solved the new framework by block coordinate descent(BCD) methods[1]. Others proposed novel alternating minimization methods and proved their convergence results. For instance, Carreira and Wang proposed a method of auxiliary coordinates(MAC) method to replace a nested neural network with a constrained problem without nesting[4]. Lin and Yao, and Lau et al proposed a BCD algorithm and proved its convergence by the Kurdyka-ojasiewicz (KL) property[12, 11]. Choromanska et al. proposed a BCD algorithm for training deep feedforward neural networks by employing the concept of co-activation memory[5]. A BCD algorithm with R-linear convergence was proposed by Zhang and Brand to train Tikhonov regularized deep neural networks[24].

3 The DLAM algorithm

In this section, we present our novel DLAM algorithm. Specifically, Section 3.1 specifies a novel formulation. Section 3.2 presents the DLAM algorithm and quadratic approximation technique to solve all subproblems.

3.1 Problem Formulation

Important notations used in this paper are shown in Table 1. Without loss of generality, we consider the simplest situation where there is only a neuron in each layer of the neural network. A typical fully-connected deep neural network

consists of L layers, each of which are defined by a linear mapping and a nonlinear mapping. The linear mapping is composed of weight vector W_l, an intercept b_l, and a nonlinear mapping is defined by activation function $h_l(\bullet)$. Given an input a_{l-1} from the $(l-1)$-th layer, the $l-th$ layer outputs $a_l = h_l(W_l a_{l-1} + b_l)$. By introducing an auxiliary variable z_l as the temporary result of the linear mapping, the deep neural network problem is formulated mathematically as follows:

Problem 1.

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

where $a_0 = x$ is the input of the deep neural network, and y is a predefined label vector. $R(z_L; y)$ is a risk function on the L-th layer and $\Omega_l(W_l)$ is a regularization term on the l-th layer. The nonlinear equality constraint $a_l = h_l(z_l)$ is the most challenging to handle here: common activation functions such as tanh and smooth sigmoid make them difficult to obtain global minimum when updating z_l[19]. To deal with this challenge, we introduce a tolerance $\varepsilon_l > 0$ and reformulate Problem 1 to the following form:

Problem 2.

$$\begin{aligned}\min_{W_l, b_l, z_l, a_l} F(\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{z}, \boldsymbol{a}) = R(z_L; y) + \sum_{l=1}^{L} \Omega_l(W_l) + \sum_{i=1}^{L} \phi(a_{l-1}, W_l, b_l, z_l)\\ + \sum_{l=1}^{L-1} \mathbb{I}(h_l(z_l) - \varepsilon_l \le a_l \le h_l(z_l) + \varepsilon_l)\end{aligned}$$

In this formulation, all notations of Problem 2 follow Problem 1. $F(\mathbf{W}, \mathbf{b}, \mathbf{z}, \mathbf{a})$ is an objective function. $\mathbf{W} = \{W_l\}_{l=1}^{L}$, $\mathbf{b} = \{b_l\}_{l=1}^{L}$, $\mathbf{z} = \{z_l\}_{l=1}^{L}$,$\mathbf{a} = \{a_l\}_{l=1}^{L-1}$. $\mathbb{I}(h_l(z_l) - \varepsilon_l \le a_l \le h_l(z_l) + \varepsilon_l)$ is an indicator function such that the value is 0 if $h_l(z_l) - \varepsilon_l \le a_l \le h_l(z_l) + \varepsilon_l$ and ∞ otherwises. The penalty term is defined as $\phi(a_{l-1}, W_l, b_l, z_l) = (\rho/2)\|z_l - W_l a_{l-1} - b_l\|_2^2$ where $\rho > 0$ a penalty parameter.

The motivation to introduce ε_l is that we project the nonlinear constraint to a convex ε_l-ball, and hence the nonconvex Problem 1 is tranformed to the multi-convex Problem 2, which is more easy to solve. Here a multi-convex problem means this problem is convex with regard to one variable while fixing others. For example, Problem 2 is convex with regard to $\mathbf{z}$ when $\mathbf{W}$, $\mathbf{b}$, and $\mathbf{a}$ are fixed. As $\rho \to \infty$ and $\varepsilon_l \to 0$, Problem 2 approaches Problem 1.

3.2 The DLAM algorithm

In this section, we present The DLAM algorithm to solve Problem 2, which is shown in Algorithm 1. Specifically, Line 4, 5, 7, and 10 update W_l, b_l,z_l and a_l, respectively. Four subproblems are discuss in detail below:

1. Update W_l

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

$$W_l^{k+1} \leftarrow \arg\min_{W_l} \phi(a_{l-1}^{k+1}, W_l, b_l^k, z_l^k) + \Omega_l(W_l)$$

Algorithm 1 the DLAM Algorithm for Solving Problem 2

Require: y, $a_0 = x$.
Ensure: $a_l(l = 1, \cdots, L-1), W_l(l = 1, \cdots, L), b_l(l = 1, \cdots, L), z_l(l = 1, \cdots, L)$.

```
Initialize ρ, k = 0.
repeat
    for l = 1 to L do
        Update W_l^{k+1} by Algorithm 2.
        Update b_l^{k+1} in equation 2.
        if l = L then
            Update z_l^{k+1} in equation 5.
        else
            Update z_l^{k+1} in equation 4.
            Update a_l^{k+1} by Algorithm 3.
        end if
    end for
    k ← k + 1.
until convergence.
Output a_l, W_l, b_l, z_l.
```

Because W_l and a_{l-1} are coupled in ϕ, solving W_l requires the inversion operation of a_{l-1}^{k+1}, which is computationally expensive. In order to handle this challenge, we define $P_l^{k+1}(W_l; \theta_l^{k+1})$ as a quadratic approximation of ϕ at W_l^k, which is mathematically reformulated as follows[2]:

$$P_l^{k+1}(W_l; \theta_l^{k+1}) = \phi(a_{l-1}^{k+1}, W_l^k, z_l^k, b_l^k) + < \nabla\phi_{W_l^k}, W_l - W_l^k > + \|\theta_l^{k+1} \circ (W_l - W_l^k)^{\circ 2}\|_{1,1}/2$$

where $\theta_l^{k+1} > 0$ is a parameter vector, $\circ$ and $\circ 2$ denote Hadamard product (elementwise product) and Hadamard power (elementwise power), respectively, $\| \bullet \|_{1,1}$ is $\ell_{1,1}$ norm. $< \bullet, \bullet >$ is a Frobenius inner product.$\nabla\phi_{W_l^k} = \rho(W_l^k a_{l-1}^{k+1} + b_l^k - z_l^k)(a_{l-1}^{k+1})^T (l = 1, \cdots, L)$ is the gradient of ϕ with regard to W_l at W_l^k. Obviously, $P_l^{k+1}(W_l^k; \theta_l^{k+1}) = \phi(a_{l-1}^{k+1}, W_l^k, b_l^k, z_l^k)$. Instead of minimizing the original subproblem, we minimize the following:

$$W_l^{k+1} \leftarrow \arg\min_{W_l} P_l^{k+1}(W_l; \theta_l^{k+1}) + \Omega_l(W_l) \tag{1}$$

For $\Omega_l(W_l)$, common regularization terms like ℓ_1 or ℓ_2 regularization lead to closed-form solutions. As for the choice of θ_l^{k+1}, the backtracking algorithm is shown in Algorithm 2. Specifically, for a given θ_l^{k+1}, we minimize equation 1 to obtain W_l^{k+1} until the condition in Line 3 is satisfied. The backtracking algorithm always terminates because as $\theta_l^{k+1} \to \infty$, $W_l^{k+1} \to W_l^k$, and W_l^k satisfies the condition in Line 3.

2. Update b_l

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

$$b_l^{k+1} \leftarrow \arg\min_{b_l} \phi(a_{l-1}^{k+1}, W_l^{k+1}, b_l, z_l^k).$$

Algorithm 2 The Backtracking Algorithm to update W_l^{k+1}

Require: a_{l-1}^{k+1}, W_l^k, b_l^k, z_l^k, ρ, some constant $\gamma > 1$ and a constant ν.
Ensure: θ_l^{k+1},W_l^{k+1}.
1: Initialize $\alpha = \nu$.
2: update ζ in equation 1 where $\theta_l^{k+1} = \alpha$.
3: **while** $\phi(a_{l-1}^{k+1}, \zeta, b_l^k, z_l^k) > P_l^{k+1}(\zeta; \alpha)$ **do**
4: $\alpha \leftarrow \alpha\gamma$.
5: update ζ in equation 1 where $\theta_l^{k+1} = \alpha$.
6: **end while**
7: Output $\theta_l^{k+1} \leftarrow \alpha$.
8: Output $W_l^{k+1} \leftarrow \zeta$.

The above subproblem has a closed-form solution $b_l^{k+1} = z_l^k - W_l^{k+1}a_{l-1}^{k+1}$. However, the value of b_l^{k+1} is subject to fluctuation as the signs of either W_l^{k+1} or a_{l-1}^{k+1} may change, which slow down the convergence of b_l^{k+1}. Therefore, we define $U_l^{k+1}(b_l; L_b)$ as a quadratic approximation of ϕ at b_l^k, which is formulated mathematically as follows[2]:

$$U_l^{k+1}(b_l; L_b) = \phi(a_{l-1}^{k+1}, W_l^{k+1}, b_l^k, z_l^k) + (\nabla\phi_{b_l^k})^T(b_l - b_l^k) + (L_b/2)\|b_l - b_l^k\|_2^2.$$

where $L_b \geq \rho$ is a parameter and $\nabla\phi_{b_l^k} = \rho(b_l^k + W_l^{k+1}a_{l-1}^{k+1} - z_l^k)$. Here $L_b \geq \rho$ is required for convergence analysis [2]. Without loss of generality, we set $L_b = \rho$. Therefore, we solve the following subproblem as follows:

$$b_l^{k+1} \leftarrow \arg\min_{b_l} U_l^{k+1}(b_l; \rho) \tag{2}$$

The solution to equation 2 is

$$b_l^{k+1} \leftarrow b_l^k - \nabla\phi_{b_l^k}/\rho \tag{3}$$

It indicates that b_l^{k+1} is closely related to b_l^k and more resistant to the sign change of W_l^{k+1} and a_{l-1}^{k+1}.

3. Update z_l

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

$$z_l^{k+1} \leftarrow \arg\min_{z_l} \phi(a_{l-1}^{k+1}, W_l^{k+1}, b_l^{k+1}, z_l) + \mathbb{I}(h_l(z_l) - \varepsilon_l \leq a_l^k \leq h_l(z_l) + \varepsilon_l)(l < L)$$
$$z_L^{k+1} \leftarrow \arg\min_{z_L} \phi(a_{L-1}^{k+1}, W_L^{k+1}, b_L^{k+1}, z_L) + R(z_L; y)$$

Due to the same reason when updating b_l, we define $V_l^{k+1}(z_l; L_z)$ as a quadratic approximation of ϕ at z_l^k, which is formulated mathematically as follows:

$$V_l^{k+1}(z_l; L_z) = \phi(a_{l-1}^{k+1}, W_l^{k+1}, b_l^{k+1}, z_l^k) + (\nabla\phi_{z_l^k})^T(z_l - z_l^k) + (L_z/2)\|z_l - z_l^k\|_2^2$$

where $L_z \geq \rho$ is a parameter and $\nabla\phi_{z_l^k} = \rho(z_l^k - W_l^{k+1}a_{l-1}^{k+1} - b_l^{k+1})$. Without loss of generality, we set $L_z = \rho$. Obviously, $V_l^{k+1}(z_l^k; \rho) = \phi(a_{l-1}^{k+1}, W_l^{k+1}, b_l^{k+1}, z_l^k)$.

Therefore, we solve the following problems:

$$z_l^{k+1} \leftarrow \arg\min_{z_l} V_l^{k+1}(z_l;\rho)+\mathbb{I}(h_l(z_l)-\varepsilon_l \le a_l^k \le h_l(z_l)+\varepsilon_l)(l<L) \quad (4)$$

$$z_L^{k+1} \leftarrow \arg\min_{z_L} V_L^{k+1}(z_L;\rho) + R(z_L;y) \quad (5)$$

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

$$z_l^{k+1} \leftarrow \min(\max(B_1^{k+1}, z_l^k - \nabla\phi_{z_l^k}/\rho), B_2^{k+1}).$$

where B_1^{k+1} and B_2^{k+1} represent the lower bound and the upper bound of the set $\{z_l|h_l(z_l)-\varepsilon_l \le a_l^k \le h_l(z_l)+\varepsilon_l\}$. equation 5 is easy to solve by Fast Iterative Soft Thresholding Algorithm(FISTA)[2].

4. Update a_l

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

$$a_l^{k+1} \leftarrow \arg\min_{a_l} \phi(a_l, W_{l+1}^k, b_{l+1}^k, z_{l+1}^k) + \mathbb{I}(h_l(z_l^{k+1}) - \varepsilon_l \le a_l \le h_l(z_l^{k+1}) + \varepsilon_l)$$

Due to the same reason when solving W_l^{k+1}, the quadratic approximation of ϕ at a_l^k is defined as

$$Q_l^{k+1}(a_l;\tau_l^{k+1}) = \phi(a_l^k, W_{l+1}^k, b_{l+1}^k, z_{l+1}^k) + \nabla\phi_{a_l^k}^T(a_l - a_l^k) + \|\tau_l^{k+1} \circ (a_l - a_l^k)^{\circ 2}\|_{1,1}/2$$

and we solve the following problem instead:

$$a_l^{k+1} \leftarrow \arg\min_{a_l} Q_l^{k+1}(a_l;\tau_l^{k+1}) + \mathbb{I}(h_l(z_l^{k+1}) - \varepsilon_l \le a_l \le h_l(z_l^{k+1}) + \varepsilon_l) \quad (6)$$

where $\tau_l^{k+1} > 0$ is a parameter vector, $\circ$ and $\circ 2$ denote Hadamard product (elementwise product) and Hadamard power(elementwise power), respectively. $\| \bullet \|_{1,1}$ a $\ell_{1,1}$ norm. $\nabla\phi_{a_l^k} = \rho(W_{l+1}^k)^T(W_{l+1}^k a_l^k + b_{l+1}^k - z_{l+1}^k)(l=1,\cdots,L-1)$ is the gradient of ϕ with regard to a_l at a_l^k. Obviously, $Q_l^{k+1}(a_l^k;\tau_l^{k+1}) = \phi(a_l^k, W_{l+1}^k, b_{l+1}^k, z_{l+1}^k)$. Because $Q_l^{k+1}(a_l;\tau_l^{k+1})$ is a quadratic function with respect to a_l, the solution can be obtained by

$$a_l^{k+1} \leftarrow a_l^k - \nabla\phi_{a_l^k}/\tau_l^{k+1}$$

given a suitable τ_l^{k+1}. Now the main focus is how to choose τ_l^{k+1}. Similar to Algorithm 2, the backtracking algorithm of finding a suitable τ_l^{k+1} is shown in Algorithm 3.

4 Convergence Analysis

In this section, we present assumptions and main convergence results of the DLAM algorithm. Specifically, Section 4.1 introduces necessary assumptions to guarantee convergence. Main convergence properties of our DLAM algorithm are presented in Section 4.2.

Algorithm 3 The Backtracking Algorithm to update a_l^{k+1}

Require: a_l^k, W_{l+1}^k, z_l^{k+1}, z_{l+1}^k, b_{l+1}^k, ρ, some constant $\eta > 1$ and μ.
Ensure: τ_l^{k+1},a_l^{k+1}.

1: Pick up $t = \mu$ such that $\beta = a_l^k - \nabla\phi_{a_l^k}/t$ and $h_l(z_l^{k+1}) - \varepsilon_l \leq \beta \leq h_l(z_l^{k+1}) + \varepsilon_l$.
2: **while** $\phi(\beta, W_{l+1}^k, z_{l+1}^k, b_{l+1}^k) > Q_l^{k+1}(\beta; t)$ **do**
3: $\quad t \leftarrow t\eta$.
4: $\quad \beta \leftarrow a_l^k - \nabla\phi_{a_l^k}/t$.
5: **end while**
6: Output $\tau_l^{k+1} \leftarrow t$.
7: Output $a_l^{k+1} \leftarrow \beta$.

4.1 Assumptions

Firstly, we recall back the concept of coercivity as follows[22]:

Definition 1 (Coercivity)**.** *Suppose $f(x_1, \cdots, x_m)$ is a function with respect to $(x_1, \cdots, x_m) \in G$ where G is the domain, then $f(x_1, \cdots, x_m)$ is coercive if $(x_1, \cdots, x_m) \in G$ and $\|(x_1, \cdots, x_m)\| \to \infty$ leads to $f \to \infty$.*

Then we propose a new concept called multi-coercitvity following the definition of coercivity, which is defined below:

Definition 2 (Multi-coercivity)**.** *Suppose $f(x_1, \cdots, x_m)$ is a function with respect to $(x_1, \cdots, x_m) \in G$ where G is the domain, then $f(x_1, \cdots, x_m)$ is coercive with regard to x_1 if $(x_1, \cdots, x_m) \in G$ and $\|x_1\| \to \infty$ while fixing $x_i (i = 2, 3, \cdots, m)$ leads to $f \to \infty$. If f is coercive with regard to all variables $x_i (i = 1, 2, \cdots, m)$, then f is multi-coercive.*

It is easy to check that multi-coercivity is an extension of coercivity. A coercive function must be multi-coercive, but not vice versa. One counterexample is $f_1(x, y) = x + y$ where f_1 is coercive with regard to x and y and hence it is multi-coercive. However, f_1 is not coercive because when $\|(x, y)\| \to \infty$ and (x, y) follows the line $x + y = 0$, then $f_1 = 0$. Given the definition of multi-coercivity, we have the following assumption:

Assumption 1 (Multi-coercivity)**.** *$F(\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{z}, \boldsymbol{a})$ is multi-coercive over the set $S = \{(\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{z}, \boldsymbol{a}) : h_l(z_l) - \varepsilon_l \leq a_l \leq h_l(z_l) + \varepsilon_l (l = 1 \cdots, L-1)\}$.*

It is easy to check that common loss functions of neural networks such as cross entropy and square loss are multi-coercive to ensure the multi-coercivity of F. Noticeably, We do not require F to be coercive. As a result, Assumption 1 is a mild condition. The next assumption guarantees all subproblems have global minima.

Assumption 2 (Subproblem Optimality)**.** *$R(z_L; y)$ and $\Omega_l(W_l)(l = 1, \cdots, L)$ are proper closed convex functions.*

Assumption 2 guarantees that every subproblem is convex and hence contains global minimum. Furthermore, Assumption 2 is also mild because common loss functions like cross entropy and square loss satisfy this assumption. At the same time, Assumption 2 also holds for common regularization terms like ℓ_1 penalty and ℓ_2 penalty.

Before stating the final assumption, we recall back the definition of quasilinearity[3] as follows:

Definition 3. *A function $f(x)$ is quasiconvex if for any sublevel set $S_\alpha(f) = \{x|f(x) \leq \alpha\}$ is a convex set. Likewise, A function $f(x)$ is quasiconcave if for any superlevel set $S_\alpha(f) = \{x|f(x) \geq \alpha\}$ is a convex set. A function $f(x)$ is quasilinear if it is both quasiconvex and quasiconcave.*

Given the definition of quasilinearity, we have the following assumption:

Assumption 3 (Quasilinearity)**.** *Activation functions $h_l(z_l)(l = 1, \cdots, n)$ are quasilinear functions.*

Assumption 3 ensures that the nonlinear constraint $a_l = h_l(z_l)$ in Problem 1 is projected in a convex set. Fortunately, almost all common nonlinear activation functions such as tanh, smooth sigmoid, and rectified linear unit(Relu) are quasilinear. Therefore, they enjoy the convergence properties stated below.

4.2 Key Convergence Results

We introduce main convergence properties of the DLAM algorithm in this section. If Assumptions 1-3 are satisfied, then Properties 1-3 stated below hold. These three properties are keystones to show the theoretical merits of the DLAM algorithm. The proofs of them are in the appendices. Finally, we present the global convergence of the DLAM based on Properties 1-3, as stated in Theorem 1. Three convergence properties are shown below:

Property 1 (Boundness)**.** *For any $\rho > 0$ and $\varepsilon_l > 0$, starting from any $(\boldsymbol{W}^0, \boldsymbol{b}^0, \boldsymbol{z}^0, \boldsymbol{a}^0)$ such that $h_l(z_l^0) - \varepsilon_l \leq a_l^0 \leq h_l(z_l^0) + \varepsilon_l (l = 1, \cdots, L)$, $\{(\boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{a}^k)\}$ is bounded, and $F(\boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{a}^k)$ is lower bounded.*

Property 1 guarantees that all variables and objective functions during iterations are bounded. The proof of Property 1 requires Lemma 3 and Assumption 1, and the proof is elaborated in Appendix B.

Property 2 (Sufficient Descent)**.** *For any $\rho > 0$ and $\varepsilon_l > 0$, we have*

$$
\begin{aligned}
&F(\boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{a}^k) - F(\boldsymbol{W}^{k+1}, \boldsymbol{b}^{k+1}, \boldsymbol{z}^{k+1}, \boldsymbol{a}^{k+1}) \\
&\geq \sum\nolimits_{l=1}^{L} \|\theta_l^{k+1} \circ (W_l^{k+1} - W_l^k)^{\circ 2}\|_{1,1}/2 + (\rho/2) \sum\nolimits_{l=1}^{L} \|b_l^{k+1} - b_l^k\|_2^2 + (\rho/2) \sum\nolimits_{l=1}^{L} \|z_l^{k+1} - z_l^k\|_2^2 \\
&+ \sum\nolimits_{l=1}^{L-1} \|\tau_l^{k+1} \circ (a_l^{k+1} - a_l^k)^{\circ 2}\|_{1,1}/2 \qquad (7)
\end{aligned}
$$

Property 2 depicts the monotonic decrease of the objective value during iterations. The proof of Property 2 is detailed in Appendix C and requires Lemma 3.

Property 3 (Subgradient Bound)**.** *There exists a constant* $C^{k+1} > 0$ *and* $g \in \partial F(\boldsymbol{W}^{k+1}, \boldsymbol{b}^{k+1}, \boldsymbol{z}^{k+1}, \boldsymbol{a}^{k+1})$ *such that*

$$\|g\| \leq C^{k+1}(\|\boldsymbol{W}^{k+1} - \boldsymbol{W}^{k}\| + \|\boldsymbol{b}^{k+1} - \boldsymbol{b}^{k}\| + \|\boldsymbol{z}^{k+1} - \boldsymbol{z}^{k}\| + \|\boldsymbol{a}^{k+1} - \boldsymbol{a}^{k}\|) \quad (8)$$

Property 3 ensures that the subgradient of the objective function is bounded by variables. The proof of Property 3 requires Property 1 and the proof process is elaborated in Appendix D. The global convergence of the DLAM algorithm is presented in the below theorem.

Theorem 1. *For any* $\rho > 0$ *and* $\varepsilon_l > 0$*, suppose a sequence* $(\boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{a}^k)$ *satisfies Assumption 1, 2 and 3. Then, starting from any* $(\boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{a}^k)$ *such that* $h_l(z_l^k) - \varepsilon_l \leq a_l^k \leq h_l(z_l^k) + \varepsilon_l (l = 1, \cdots, L)$*, the sequence has at least a limit point* $(\boldsymbol{W}^*, \boldsymbol{b}^*, \boldsymbol{z}^*, \boldsymbol{a}^*)$*, and any limit point* $(\boldsymbol{W}^*, \boldsymbol{b}^*, \boldsymbol{z}^*, \boldsymbol{a}^*)$ *is a stationary solution of the objective function* $F(\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{z}, \boldsymbol{a})$ *defined in Problem 2. That is,* $0 \in \partial F(\boldsymbol{W}^*, \boldsymbol{b}^*, \boldsymbol{z}^*, \boldsymbol{a}^*)$*. Moreover, if* F *is a Kurdyka-Lojasiewicz (KL) function, then* $(\boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{a}^k)$ *converges globally to a unique point* $(\boldsymbol{W}^*, \boldsymbol{b}^*, \boldsymbol{z}^*, \boldsymbol{a}^*)$*. The worst convergence rate of* $\|\boldsymbol{W}^{k+1} - \boldsymbol{W}^k\|_2^2 + \|\boldsymbol{b}^{k+1} - \boldsymbol{b}^k\|_2^2 + \|\boldsymbol{z}^{k+1} - \boldsymbol{z}^k\|_2^2 + \|\boldsymbol{a}^{k+1} - \boldsymbol{a}^k\|_2^2$ *is* $o(1/k)$*.*

Proof. Because Assumption 1, 2 and 3 are satisfied, Property 1, 2 and 3 hold. Since $(\mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{a}^k)$ is bounded, there exists a subsequence $(\mathbf{W}^s, \mathbf{b}^s, \mathbf{z}^s, \mathbf{a}^s)$ such that $(\mathbf{W}^s, \mathbf{b}^s, \mathbf{z}^s, \mathbf{a}^s) \to (\mathbf{W}^*, \mathbf{b}^*, \mathbf{z}^*, \mathbf{a}^*)$ where $(\mathbf{W}^*, \mathbf{b}^*, \mathbf{z}^*, \mathbf{a}^*)$ is a limit point. By Property 1 and 2, $F(\mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{a}^k)$ is non-increasing and lower bounded and hence converged. By Property 2, we prove that $\|\mathbf{W}^{k+1} - \mathbf{W}^k\| \to 0$, $\|\mathbf{b}^{k+1} - \mathbf{b}^k\| \to 0$, $\|\mathbf{z}^{k+1} - \mathbf{z}^k\| \to 0$ and $\|\mathbf{a}^{k+1} - \mathbf{a}^k\| \to 0$ as $k \to \infty$. We infer there exists $g^k \in \partial F(\mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{a}^k)$ such that $\|g^k\| \to 0$ as $k \to \infty$ based on Property 3. Specifically, $\|g^s\| \to 0$ as $s \to \infty$. According to the definition of general subgradient (Defintion 8.3 in[16]), we have $0 \in \partial F(\mathbf{W}^*, \mathbf{b}^*, \mathbf{z}^*, \mathbf{a}^*)$. In other words, the limit point $(\mathbf{W}^*, \mathbf{b}^*, \mathbf{z}^*, \mathbf{a}^*)$ is a stationary solution of F. Similar to the proof of Proposition 3.1 in[22], if F is a KL function, the global convergence to a unique limit point is achieved. The worst convergence rate of $\|\mathbf{W}^{k+1} - \mathbf{W}^k\|_2^2 + \|\mathbf{b}^{k+1} - \mathbf{b}^k\|_2^2 + \|\mathbf{z}^{k+1} - \mathbf{z}^k\|_2^2 + \|\mathbf{a}^{k+1} - \mathbf{a}^k\|_2^2$ is derived directly from Lemma 1.2 in[7]. □

Theorem 1 shows that our DLAM algorithm converges globally no matter what ρ and ε_l we choose. Therefore, our DLAM algorithm is parameter-restriction free, namely, the choice of parameters has no effect on the convergence of our DLAM algorithm.

5 Experiment

In this section, we evaluate the DLAM using two common deep learning datasets: MNIST and Fashion-MNIST. We compare our DLAM method with the state-of-the-art ADMM method by[20] on both datasets with the same settings. And all experiments were conducted on 64-bit Ubuntu16.04 LTS with Intel(R) Xeon processor and GTX1080 GPU support.

5.1 Experiment Setup

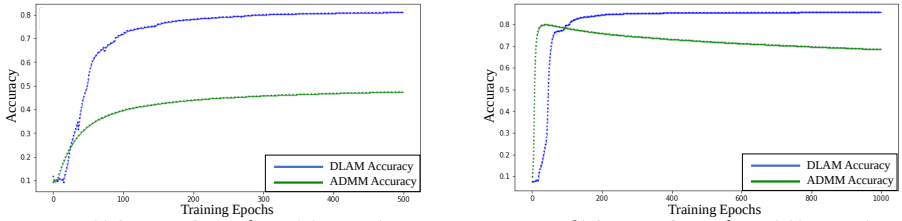
For neural network structure, we choose a three-layer full connected neural network. We also compare DLAM and ADMM's performance with the different number of neurons (10 or 100 neurons) in each layer. We use two datasets: MNIST and Fashion-MNIST. MNIST is a hand-written digit (0-9) image dataset, which contains 60,000 training images and 10,000 testing images with 28x28 pixels. Fashion-MNIST also consists of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes, e.g. T-shirt, Bag, etc.

5.2 Performance Comparison

First, we show the performance comparison between the DLAM and the ADMM on the MNIST dataset. We use two network sizes as mentioned: 3-layer Multi-Layer Perceptron (MLP) with 10 neurons per layer or 100 neurons per layer. The training accuracy curves on MNIST are shown in Fig. 1. By comparison, we could see that DLAM outperforms ADMM under both settings. Further, when we use MLP with 10 neurons per layer, ADMM could only achieve 50% training accuracy on MNIST, while DLAM could achieve about 82% accuracy. When using MLP with 100 neurons per layer, ADMM could also achieve about 80% accuracy but DLAM could achieve about 85% at this settings. In addition, with the training iteration, ADMM gradually diverges with slightly decreasing accuracy, while DLAM converges well with no such phenomenon.

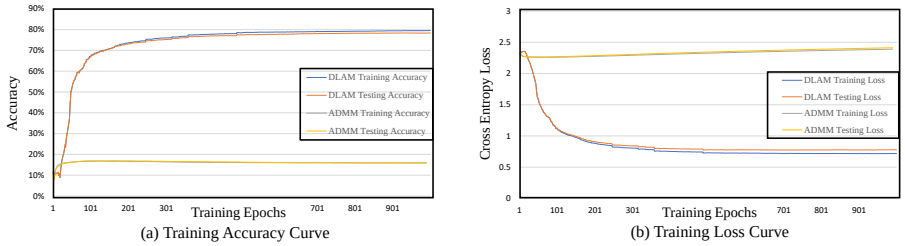
Then we show the performance comparison on Fashion-MNIST. Fig. 3 Fig. 3 shows the training/testing accuracy and loss on MLP with 10 neurons per layer. On both training set and testing set, as shown in Fig. 3 (a), DLAM could achieve near 80% accuracy while ADMM fails to achieve 20% accuracy on both training and testing datasets (Performance on training and testing are nearly the same with no overfitting). This is also shown and proved by the training/testing loss curve in Fig. 3, in which DLAM's loss could decrease to $\sim$0.7 while ADMM stops at about $\sim$2.3.

We further use another MLP structure with 100 neurons per layer and compare DLAM and ADMM's performance on it. The experiment results are shown in Fig. 3. A similar conclusion could be drawn that our DLAM achieves much better performance than ADMM.



(a) Accuracy Curve of MLP(10 neurons) (b) Accuracy Curve of MLP(100 neurons)

Figure 1: Training Accuracy curves of DLAM and ADMM during training phases on MNIST for a 3-layer MLP with 10 neurons (a) or 100 neurons (b) per layer.



(a) Training Accuracy Curve (b) Training Loss Curve

Figure 2: Training Accuracy and Loss curves of DLAM and ADMM during training phases on Fashion-MNIST (3-layer MLP with 10 neurons per layer).

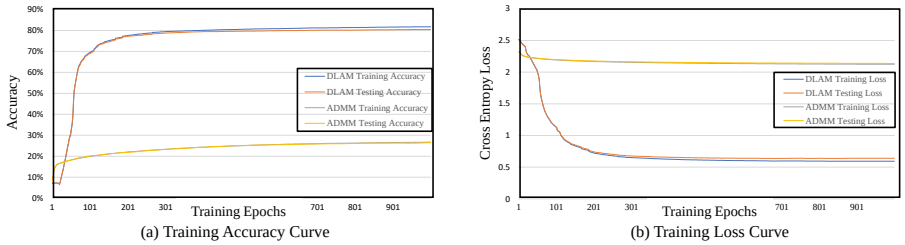
Combining MNIST and Fashion-MNIST performance, DLAM shows consistent better performance than ADMM on both datasets. ADMM achieves good accuracy on MNIST but its performance on Fashion-MNIST degrades significantly.

6 Conclusion

Even though stochastic gradient descent(SGD) is a priority to train deep neural networks, alternating minimization methods have attracted the attention of researchers becuase they have several advantages including solid theoretical guarantees and avoiding gradient vanishing problems. In this paper, we propose a novel formulation of the original deep neural network problem and a novel deep learning alternating minimization(DLAM) algorithm. Specifically, the nonlinear constraint is projected into a convex set so that all subproblems are solvable. At the same time, the quadratic approximation technique and the backtracking algroithm are applied to avoid the matrix inversion. Furthermore, several mild assumptions are estiablished to prove the global convergence of our DLAM algorithm. Experiments on real-world datasets demonstrate the effectiveness of our DLAM algorithm.

References

[1] Armin Askari, Geoffrey Negiar, Rajiv Sambharya, and Laurent El Ghaoui. Lifted neural networks. *arXiv preprint arXiv:1805.01532*, 2018.



(a) Training Accuracy Curve (b) Training Loss Curve

Figure 3: Training Accuracy and Loss curves of DLAM and ADMM during training phases on Fashion-MNIST (3-layer MLP with 100 neurons per layer).

[2] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[4] Miguel Carreira-Perpinan and Weiran Wang. Distributed optimization of deeply nested systems. In *Artificial Intelligence and Statistics*, pages 10–19, 2014.

[5] Anna Choromanska, Sadhana Kumaravel, Ronny Luss, Irina Rish, Brian Kingsbury, Ravi Tejwani, and Djallel Bouneffouf. Beyond backprop: Alternating minimization with co-activation memory. *arXiv preprint arXiv:1806.09077*, 2018.

[6] 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.

[7] Wei Deng, Ming-Jun Lai, Zhimin Peng, and Wotao Yin. Parallel multiblock admm with o (1/k) convergence. *Journal of Scientific Computing*, 71(2):712–736, 2017.

[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] Tim Tsz-Kit Lau Shaobo Lin Yuan Yao Jinshan Zeng, Shikang Ouyang. Global convergence in deep learning with variable splitting via the kurdyka-ojasiewicz property. *arXiv preprint arXiv:1803.00225*, 2018.

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[11] Tim Tsz-Kit Lau, Jinshan Zeng, Baoyuan Wu, and Yuan Yao. A proximal block coordinate descent algorithm for deep neural network training. 2018.

[12] Tianyi Lin, Shiqian Ma, and Shuzhong Zhang. Global convergence of unmodified 3-block admm for a class of convex minimization problems. *arXiv preprint arXiv:1505.04252*, 2015.

[13] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[14] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

[15] Herbert Robbins and S Monro. [a]a stochastic approximation method, [o] annals math. *Statistics*, 22:400–407, 1951.

[16] R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

[17] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.

[18] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

[19] 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.

[20] 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.

[21] T Tieleman and G Hinton. Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. Technical report, Technical Report. Available online: https://zh. coursera. org/learn/neuralnetworks/lecture/YQHki/rmsprop-divide-the-gradient-by-a-running-average-of-its-recent-magnitude (accessed on 21 April 2017).

[22] Yu Wang, Wotao Yin, and Jinshan Zeng. Global convergence of admm in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, pages 1–35, 2015.

[23] Guoqiang Zhang and W Bastiaan Kleijn. Training deep neural networks via optimization over graphs. *arXiv preprint arXiv:1702.03380*, 2017.

[24] 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.

[25] Ziming Zhang, Yuting Chen, and Venkatesh Saligrama. Efficient training of very deep neural networks for supervised hashing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1487–1495, 2016.

Appendix

A Lemmas for the Proofs of Properties

The proofs of Lemma 1 and Lemma 2 both require Assumptions 2 and 3. The proof of Lemma 3 requires Lemma 1 and Lemma 2. To simplify the notation, $\mathbf{W}_{\leq l}^{k+1} = \{\{W_i^{k+1}\}_{i=1}^{l}, \{W_i^k\}_{i=l+1}^{L}\}$, $\mathbf{b}_{\leq l}^{k+1} = \{\{b_i^{k+1}\}_{i=1}^{l}, \{b_i^k\}_{i=l+1}^{L}\}$, $\mathbf{z}_{\leq l}^{k+1} = \{\{z_i^{k+1}\}_{i=1}^{l}, \{z_i^k\}_{i=l+1}^{L}\}$ and $\mathbf{a}_{\leq l}^{k+1} = \{\{a_i^{k+1}\}_{i=1}^{l}, \{a_i^k\}_{i=l+1}^{L-1}\}$. The following several lemmas are preliminary results.

Lemma 1. *equation 1 holds if and only if there exists $s \in \partial\Omega_l(W_l^{k+1})$, the subgradient of $\Omega_l(W_l^{k+1})$ such that*

$$\nabla\phi_{W_l^k} + \theta_l^{k+1} \circ (W_l^{k+1} - W_l^k) + s = 0$$

Likewise, equation 4 holds if and only if there exists $r \in \partial\mathbb{I}(h_l(z_l^{k+1}) - \varepsilon_l \leq a_l^k \leq h_l(z_l^{k+1}) + \varepsilon_l)$ such that

$$\nabla\phi_{z_l^k} + \rho(z_l^{k+1} - z_l^k) + r = 0$$

equation 5 holds if and only if there exists $u \in \partial R(z_L^{k+1}; y)$ such that

$$\nabla\phi_{z_L^k} + \rho(z_L^{k+1} - z_L^k) + u = 0$$

equation 6 holds if and only if there exists $v \in \partial\mathbb{I}(h_l(z_l^{k+1}) - \varepsilon_l \leq a_l^{k+1} \leq h_l(z_l^{k+1}) + \varepsilon_l)$ such that

$$\nabla\phi_{a_l^k} + \rho(a_l^{k+1} - a_l^k) + v = 0$$

Proof. These can be obtained by directly applying the optimality conditions of equation 1, equation 4, equation 5 and equation 6, respectively. □

Lemma 2. *For equation 4, equation 5 and equation 2, if $L_b \geq \rho$ and $L_z \geq \rho$,then the following inequalities hold:*

$$U_l^{k+1}(b_l^{k+1}; L_b) \geq \phi(a_{l-1}^{k+1}, W_l^{k+1}, b_l^{k+1}, z_l^k) \tag{9}$$

$$V_l^{k+1}(z_l^{k+1}; L_z) \geq \phi(a_{l-1}^{k+1}, W_l^{k+1}, b_l^{k+1}, z_l^{k+1}) \tag{10}$$

Proof. Because $\phi(a_{l-1}, W_l, b_l, z_l)$ is differentiable continuous with respect to b_l and z_l with Lipschitz coefficient ρ (the definition of Lipschitz differentiablity can be found in[2]), we directly apply Lemma 2.1 in[2] to ϕ to obtain equation 9 and equation 10, respectively. □

Lemma 3. *It holds that for* $\forall k \in \mathbb{N}$ *and* $l = 1, 2, \cdots, L$,

$$F(\boldsymbol{W}_{\leq l-1}^{k+1}, \boldsymbol{b}_{\leq l-1}^{k+1}, \boldsymbol{z}_{\leq l-1}^{k+1}, \boldsymbol{a}_{\leq l-1}^{k+1}) - F(\boldsymbol{W}_{\leq l}^{k+1}, \boldsymbol{b}_{\leq l-1}^{k+1}, \boldsymbol{z}_{\leq l-1}^{k+1}, \boldsymbol{a}_{\leq l-1}^{k+1}) \geq \|\theta_l^{k+1} \circ (W_l^{k+1} - W_l^k)^{\circ 2}\|_{1,1}/2. \tag{11}$$

$$F(\boldsymbol{W}_{\leq l}^{k+1}, \boldsymbol{b}_{\leq l-1}^{k+1}, \boldsymbol{z}_{\leq l-1}^{k+1}, \boldsymbol{a}_{\leq l-1}^{k+1}) - F(\boldsymbol{W}_{\leq l}^{k+1}, \boldsymbol{b}_{\leq l}^{k+1}, \boldsymbol{z}_{\leq l-1}^{k+1}, \boldsymbol{a}_{\leq l-1}^{k+1}) \geq (\rho/2)\|b_l^{k+1} - b_l^k\|_2^2. \tag{12}$$

$$F(\boldsymbol{W}_{\leq l}^{k+1}, \boldsymbol{b}_{\leq l}^{k+1}, \boldsymbol{z}_{\leq l-1}^{k+1}, \boldsymbol{a}_{\leq l-1}^{k+1}) - F(\boldsymbol{W}_{\leq l}^{k+1}, \boldsymbol{b}_{\leq l}^{k+1}, \boldsymbol{z}_{\leq l}^{k+1}, \boldsymbol{a}_{\leq l-1}^{k+1}) \geq (\rho/2)\|z_l^{k+1} - z_l^k\|_2^2. \tag{13}$$

$$F(\boldsymbol{W}_{\leq l}^{k+1}, \boldsymbol{b}_{\leq l}^{k+1}, \boldsymbol{z}_{\leq l}^{k+1}, \boldsymbol{a}_{\leq l-1}^{k+1}) - F(\boldsymbol{W}_{\leq l}^{k+1}, \boldsymbol{b}_{\leq l}^{k+1}, \boldsymbol{z}_{\leq l}^{k+1}, \boldsymbol{a}_{\leq l}^{k+1}) \geq \|\tau_l^{k+1} \circ (a_l^{k+1} - a_l^k)^{\circ 2}\|_{1,1}/2. \tag{14}$$

Proof. Essentially, all inequalities can be obtained by applying optimality conditions of updating W_l^{k+1}, b_l^{k+1}, z_l^{k+1} and a_l^{k+1}, respectively. We only prove equation 11 and equation 13 since equation 14 and equation 12 follow the same routine of equation 11 and equation 13, respectively.

Firstly, we focus on proving equation 11. The stopping criterion of Algorithm 2 shows that

$$\phi(a_{l-1}^{k+1}, W_l^{k+1}, b_l^k, z_l^k) \leq P_l^{k+1}(W_l^{k+1}; \theta_l^{k+1}). \tag{15}$$

Because $\Omega_{W_l}(W_l)$ is convex, according to the definition of subgradient, we have

$$\Omega_l(W_l^k) \geq \Omega_l(W_l^{k+1}) + s^T(W_l^k - W_l^{k+1}) \tag{16}$$

where s is defined in the premise of Lemma 1. Therefore, we have

$$\begin{aligned}
&F(\mathbf{W}_{\leq l-1}^{k+1}, \mathbf{b}_{\leq l-1}^{k+1}, \mathbf{z}_{\leq l-1}^{k+1}, \mathbf{a}_{\leq l-1}^{k+1}) - F(\mathbf{W}_{\leq l}^{k+1}, \mathbf{b}_{\leq l-1}^{k+1}, \mathbf{z}_{\leq l-1}^{k+1}, \mathbf{a}_{\leq l-1}^{k+1}) \\
&= \phi(a_{l-1}^{k+1}, W_l^k, b_l^k, z_l^k) + \Omega_l(W_l^k) - \phi(a_{l-1}^{k+1}, W_l^{k+1}, b_l^k, z_l^k) - \Omega_l(W_l^{k+1}) \text{ (Definition of } F \text{ in Problem 2)} \\
&\geq \Omega_l(W_l^k) - \Omega_l(W_l^{k+1}) - \nabla\phi_{W_l^k}^T(W_l^{k+1} - W_l^k) - \|\theta_l^{k+1} \circ (W_l^{k+1} - W_l^k)^{\circ 2}\|_{1,1}/2 \\
&\text{(equation 15)} \\
&\geq s^T(W_l^k - W_l^{k+1}) - \nabla\phi_{W_l^k}^T(W_l^{k+1} - W_l^k) - \|\theta_l^{k+1} \circ (W_l^{k+1} - W_l^k)^{\circ 2}\|_{1,1}/2 \\
&\text{(equation 16)} \\
&= (s + \nabla\phi_{W_l^k}^T)(W_l^k - W_l^{k+1}) - \|\theta_l^{k+1} \circ (W_l^{k+1} - W_l^k)^{\circ 2}\|_{1,1}/2 \\
&= \|\theta_l^{k+1} \circ (W_l^{k+1} - W_l^k)^{\circ 2}\|_{1,1}/2 \text{ (Lemma 1).}
\end{aligned}$$

Secondly, we focus on proving equation 13. For $l < L$, because $\mathbb{I}(h_l(z_l) - \varepsilon_l \leq a_l^k \leq h_l(z_l) + \varepsilon_l)$ is convex with regard to z_l, according to the definition of subgradient, we have

$$\mathbb{I}(h_l(z_l^k) - \varepsilon_l \leq a_l^k \leq h_l(z_l^k) + \varepsilon_l) \geq \mathbb{I}(h_l(z_l^{k+1}) - \varepsilon_l \leq a_l^k \leq h_l(z_l^{k+1}) + \varepsilon_l) + r^T(z_l^k - z_l^{k+1}) \tag{17}$$

where r is defined in Lemma 1.

$$
\begin{aligned}
&F(\mathbf{W}_{\leq l}^{k+1}, \mathbf{b}_{\leq l}^{k+1}, \mathbf{z}_{\leq l-1}^{k+1}, \mathbf{a}_{\leq l-1}^{k+1}) - F(\mathbf{W}_{\leq l}^{k+1}, \mathbf{b}_{\leq l}^{k+1}, \mathbf{z}_{\leq l}^{k+1}, \mathbf{a}_{\leq l-1}^{k+1}) \\
&= \phi(a_{l-1}^{k+1}, W_l^{k+1}, b_l^{k+1}, z_l^k) + \mathbb{I}(h_l(z_l^k) - \varepsilon_l \leq a_l^k \leq h_l(z_l^k) + \varepsilon_l) - \phi(a_{l-1}^{k+1}, W_l^{k+1}, b_l^{k+1}, z_l^{k+1}) \\
&- \mathbb{I}(h_l(z_l^{k+1}) - \varepsilon_l \leq a_l^k \leq h_l(z_l^{k+1}) + \varepsilon_l) (\text{Definition of } F \text{ in Problem 2}) \\
&\geq -\nabla \phi_{z_l^k}^T (z_l^{k+1} - z_l^k) - (\rho/2)\|z_l^{k+1} - z_l^k\|_2^2 + \mathbb{I}(h_l(z_l^k) - \varepsilon_l \leq a_l^k \leq h_l(z_l^k) + \varepsilon_l) \\
&- \mathbb{I}(h_l(z_l^{k+1}) - \varepsilon_l \leq a_l^k \leq h_l(z_l^{k+1}) + \varepsilon_l) (\text{equation 10}) \\
&\geq -\nabla \phi_{z_l^k}^T (z_l^{k+1} - z_l^k) - (\rho/2)\|z_l^{k+1} - z_l^k\|_2^2 + r^T(z_l^k - z_l^{k+1}) (\text{equation 17}) \\
&= -\nabla \phi_{z_l^k}^T (z_l^{k+1} - z_l^k) - (\rho/2)\|z_l^{k+1} - z_l^k\|_2^2 + (\nabla \phi_{z_l^k} + \rho(z_l^{k+1} - z_l^k))^T (z_l^{k+1} - z_l^k) (\text{Lemma 1}) \\
&= (\rho/2)\|z_l^{k+1} - z_l^k\|_2^2.
\end{aligned}
$$

For z_L, the same routine applies. □

B Proof of Property 1.

Proof. The lower boundness of $F(\mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{a}^k)$ can be directly obtained by Lemma 3. Moreover, we set $l = L$ and $k := k + 1$ in equation 13 to get

$$F(\mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{a}^k) \leq F(\mathbf{W}^k, \mathbf{b}^k, \mathbf{z}_{\leq L-1}^k, \mathbf{a}^k) - (\rho/2)\|z_L^k - z_L^{k-1}\|_2^2.$$

Because $F(\mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{a}^k)$ is multi-coercive and $F(\mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{a}^k)$ is upper bounded by $F(\mathbf{W}^k, \mathbf{b}^k, \mathbf{z}_{\leq L-1}^k, \mathbf{a}^k) - (\rho/2)\|z_L^k - z_L^{k-1}\|_2^2$, $(\mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{a}^k)$ is also bounded. □

C Proof of Property 2.

Proof. This can be obtained by adding equation 11, equation 12 and equation 13 from $l = 1$ to $l = L$ and equation 14 from $l = 1$ to $l = L - 1$. □

D Proof of Property 3.

Proof. We know that

$$\partial F(\mathbf{W}_1^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{a}^{k+1}) = (\partial F_{\mathbf{W}^{k+1}}, \nabla F_{\mathbf{b}^{k+1}}, \partial F_{\mathbf{z}^{k+1}}, \partial F_{\mathbf{a}^{k+1}})$$

where $\partial F_{\mathbf{W}^{k+1}} = \{\partial F_{W_l^{k+1}}\}_{l=1}^L$, $\nabla F_{\mathbf{b}^{k+1}} = \{\nabla F_{b_l^{k+1}}\}_{l=1}^L$, $\partial F_{\mathbf{z}^{k+1}} = \{\partial F_{z_l^{k+1}}\}_{l=1}^L$ and $\partial F_{\mathbf{a}^{k+1}} = \{\partial F_{a_l^{k+1}}\}_{l=1}^{L-1}$. To prove Property 3, we need to give a upper bound of $\partial F_{W_l^{k+1}}$, $\partial F_{b_l^{k+1}}$, $\partial F_{z_l^{k+1}}$ and $\partial F_{a_l^{k+1}}$ by a linear combination of $\|W_l^{k+1} - W_l^k\|$,

$\|b_l^{k+1} - b_l^k\|$, $\|z_l^{k+1} - z_l^k\|$ and $\|a_l^{k+1} - a_l^k\|$.

For W_l^{k+1},

$$\begin{aligned}
\partial F_{W_l^{k+1}} &= \partial\Omega_l(W_l^{k+1}) + \nabla\phi_{W_l^{k+1}}(a_{l-1}^{k+1}, W_l^{k+1}, b_l^{k+1}, z_l^{k+1})(\text{Definition of } F \text{ in equation 2}) \\
&= \nabla\phi_{W_l^{k+1}}(a_{l-1}^{k+1}, W_l^{k+1}, b_l^{k+1}, z_l^{k+1}) - \nabla\phi_{W_l^k}(a_{l-1}^{k+1}, W_l^k, b_l^k, z_l^k) - \theta_l^{k+1} \circ (W_l^{k+1} - W_l^k) \\
&+ \partial\Omega_l(W_l^{k+1}) + \nabla\phi_{W_l^k}(a_{l-1}^{k+1}, W_l^k, b_l^k, z_l^k) + \theta_l^{k+1} \circ (W_l^{k+1} - W_l^k) \\
&= \rho(W_l^{k+1} - W_l^k)a_{l-1}^{k+1}(a_{l-1}^{k+1})^T + \rho(b_l^{k+1} - b_l^k)(a_{l-1}^{k+1})^T - \rho(z_l^{k+1} - z_l^k)(a_{l-1}^{k+1})^T - \theta_l^{k+1} \circ (W_l^{k+1} - W_l^k) \\
&+ \partial\Omega_l(W_l^{k+1}) + \nabla\phi_{W_l^k}(a_{l-1}^{k+1}, W_l^k, b_l^k, z_l^k) + \theta_l^{k+1} \circ (W_l^{k+1} - W_l^k)
\end{aligned}$$

Because

$$\begin{aligned}
&\|\rho(W_l^{k+1} - W_l^k)a_{l-1}^{k+1}(a_{l-1}^{k+1})^T + \rho(b_l^{k+1} - b_l^k)(a_{l-1}^{k+1})^T - \rho(z_l^{k+1} - z_l^k)(a_{l-1}^{k+1})^T - \theta_l^{k+1} \circ (W_l^{k+1} - W_l^k)\| \\
&\leq \rho\|(W_l^{k+1} - W_l^k)a_{l-1}^{k+1}(a_{l-1}^{k+1})^T\| + \rho\|(b_l^{k+1} - b_l^k)(a_{l-1}^{k+1})^T\| + \rho\|(z_l^{k+1} - z_l^k)(a_{l-1}^{k+1})^T\| \\
&+ \|\theta_l^{k+1} \circ (W_l^{k+1} - W_l^k)\|(\text{triangle inequality}) \\
&\leq \rho\|W_l^{k+1} - W_l^k\|\|a_{l-1}^{k+1}\|\|a_{l-1}^{k+1}\| + \rho\|b_l^{k+1} - b_l^k\|\|a_{l-1}^{k+1}\| + \rho\|z_l^{k+1} - z_l^k\|\|a_{l-1}^{k+1}\| \\
&+ \|\theta_l^{k+1}\|\|W_l^{k+1} - W_l^k\|(\text{Cauchy-Schwarz inequality})
\end{aligned}$$

and the optimality condition of equation 1 yeilds

$$0 \in \partial\Omega_l(W_l^{k+1}) + \nabla\phi_{W_l^k}(a_{l-1}^{k+1}, W_l^k, b_l^k, z_l^k) + \theta_l^{k+1} \circ (W_l^{k+1} - W_l^k)$$

Because a_{l-1}^{k+1} is bounded by Property 1, $\|\partial F_{W_l^{k+1}}\|$ can be upper bounded by a linear combination of $\|W_l^{k+1} - W_l^k\|$, $\|b_l^{k+1} - b_l^k\|$ and $\|z_l^{k+1} - z_l^k\|$.

For b_l^{k+1},

$$\begin{aligned}
\nabla F_{b_l^{k+1}} &= \nabla\phi_{b_l^{k+1}}(a_{l-1}^{k+1}, W_{l-1}^{k+1}, b_l^{k+1}, z_l^{k+1}) \\
&= \nabla\phi_{b_l^{k+1}}(a_{l-1}^{k+1}, W_{l-1}^{k+1}, b_l^{k+1}, z_l^{k+1}) - \nabla\phi_{b_l^k}(a_{l-1}^{k+1}, W_{l-1}^{k+1}, b_l^k, z_l^k) - \rho(b_l^{k+1} - b_l^k) \\
&(\nabla\phi_{b_l^k}(a_{l-1}^{k+1}, W_{l-1}^{k+1}, b_l^k, z_l^k) + \rho(b_l^{k+1} - b_l^k)) = 0 \text{ by the optimality condition of equation 2}) \\
&= \rho(z_l^{k+1} - z_l^k).
\end{aligned}$$

Therefore, $\|\nabla F_{b_l^{k+1}}\|$ is linearly independent on $\|z_l^{k+1} - z_l^k\|$.

For $z_l^{k+1}(l < L)$,

$$\begin{aligned}
\partial F_{z_l^{k+1}} &= \nabla\phi_{z_l^{k+1}}(a_{l-1}^{k+1}, W_{l-1}^{k+1}, b_l^{k+1}, z_l^{k+1}) + \partial\mathbb{I}(h_l(z_l^{k+1}) - \varepsilon_l \leq a_l^{k+1} \leq h_l(z_l^{k+1}) + \varepsilon_l) \\
&= \nabla\phi_{z_l^{k+1}}(a_{l-1}^{k+1}, W_{l-1}^{k+1}, b_l^{k+1}, z_l^{k+1}) - \nabla\phi_{z_l^k}(a_{l-1}^{k+1}, W_{l-1}^{k+1}, b_l^{k+1}, z_l^k) - \rho(z_l^{k+1} - z_l^k) \\
&+ \partial\mathbb{I}(h_l(z_l^{k+1}) - \varepsilon_l \leq a_l^{k+1} \leq h_l(z_l^{k+1}) + \varepsilon_l) - \partial\mathbb{I}(h_l(z_l^{k+1}) - \varepsilon_l \leq a_l^k \leq h_l(z_l^{k+1}) + \varepsilon_l) \\
&+ \nabla\phi_{z_l^k}(a_{l-1}^{k+1}, W_{l-1}^{k+1}, b_l^{k+1}, z_l^k) + \partial\mathbb{I}(h_l(z_l^{k+1}) - \varepsilon_l \leq a_l^k \leq h_l(z_l^{k+1}) + \varepsilon_l) + \rho(z_l^{k+1} - z_l^k) \\
&= \nabla\phi_{z_l^{k+1}}(a_{l-1}^{k+1}, W_{l-1}^{k+1}, b_l^{k+1}, z_l^{k+1}) - \nabla\phi_{z_l^k}(a_{l-1}^{k+1}, W_{l-1}^{k+1}, b_l^{k+1}, z_l^k) - \rho(z_l^{k+1} - z_l^k) \ (0 \in \partial\mathbb{I}(\bullet) \text{ and} \\
&0 \in (\nabla\phi_{z_l^k}(a_{l-1}^{k+1}, W_{l-1}^{k+1}, b_l^{k+1}, z_l^k) + \rho(z_l^{k+1} - z_l^k) + \partial\mathbb{I}(h_l(z_l^{k+1}) - \varepsilon_l \leq a_l^k \leq h_l(z_l^{k+1}) + \varepsilon_l)) \text{ by equation} \\
&= 0
\end{aligned}$$

Likewise, we can prove $\partial F_{z_L^{k+1}} = 0$ by the optimality condition of equation 5.

For a_l^{k+1},

$$\begin{aligned}
\partial F_{a_l^{k+1}} &= \nabla\phi_{a_l^{k+1}}(a_l^{k+1}, W_{l+1}^{k+1}, z_{l+1}^{k+1}, b_{l+1}^{k+1}) + \partial\mathbb{I}(h_l(z_l^{k+1}) - \varepsilon_l \le a_l^{k+1} \le h_l(z_l^{k+1}) + \varepsilon_l) \\
&= \nabla\phi_{a_l^{k+1}}(a_l^{k+1}, W_{l+1}^{k+1}, z_{l+1}^{k+1}, b_{l+1}^{k+1}) + \partial\mathbb{I}(h_l(z_l^{k+1}) - \varepsilon_l \le a_l^{k+1} \le h_l(z_l^{k+1}) + \varepsilon_l) \\
&- \nabla\phi_{a_l^k}(a_l^k, W_{l+1}^k, z_{l+1}^k, b_{l+1}^k) - \tau_l^{k+1} \circ (a_l^{k+1} - a_l^k) \\
&+ \nabla\phi_{a_l^k}(a_l^k, W_{l+1}^k, z_{l+1}^k, b_{l+1}^k) + \tau_l^{k+1} \circ (a_l^{k+1} - a_l^k) \\
&= (W_{l+1}^{k+1})^T(W_{l+1}^{k+1}a_l^{k+1} - z_{l+1}^{k+1} - b_{l+1}^{k+1}) - (W_{l+1}^k)^T(W_{l+1}^k a_l^k - z_{l+1}^k - b_{l+1}^k) - \tau_l^{k+1} \circ (a_l^{k+1} - a_l^k) \\
&+ \nabla\phi_{a_l^k}(a_l^k, W_{l+1}^k, z_{l+1}^k, b_{l+1}^k) + \tau_l^{k+1} \circ (a_l^{k+1} - a_l^k) + \partial\mathbb{I}(h_l(z_l^{k+1}) - \varepsilon_l \le a_l^{k+1} \le h_l(z_l^{k+1}) + \varepsilon_l)
\end{aligned}$$

Because

$$\begin{aligned}
&\|(W_{l+1}^{k+1})^T(W_{l+1}^{k+1}a_l^{k+1} - z_{l+1}^{k+1} - b_{l+1}^{k+1}) - (W_{l+1}^k)^T(W_{l+1}^k a_l^k - z_{l+1}^k - b_{l+1}^k) - \tau_l^{k+1} \circ (a_l^{k+1} - a_l^k)\| \\
&= \|((W_{l+1}^{k+1})^T W_{l+1}^{k+1} a_l^{k+1} - (W_l^k)^T W_{l+1}^k a_l^k) + ((W_{l+1}^{k+1})^T z_{l+1}^{k+1} - (W_{l+1}^k)^T z_{l+1}^k) \\
&+ ((W_{l+1}^{k+1})^T b_{l+1}^{k+1} - (W_{l+1}^k)^T b_{l+1}^k) + (\tau_l^{k+1} \circ (a_l^{k+1} - a_l^k))\| \\
&\le \|(W_{l+1}^{k+1})^T W_{l+1}^{k+1} a_l^{k+1} - (W_l^k)^T W_{l+1}^k a_l^k\| + \|(W_{l+1}^{k+1})^T z_{l+1}^{k+1} - (W_{l+1}^k)^T z_{l+1}^k\| \\
&+ \|(W_{l+1}^{k+1})^T b_{l+1}^{k+1} - (W_{l+1}^k)^T b_{l+1}^k\| + \|\tau_l^{k+1} \circ (a_l^{k+1} - a_l^k)\| \text{(triangle inequality)} \\
&= \|(W_{l+1}^{k+1} - W_{l+1}^k)^T W_{l+1}^{k+1} a_l^{k+1} + (W_{l+1}^k)^T (W_{l+1}^{k+1} - W_{l+1}^k) a_l^{k+1} + (W_{l+1}^k)^T W_{l+1}^k (a_l^{k+1} - a_l^k)\| \\
&+ \|(W_{l+1}^{k+1})^T (z_{l+1}^{k+1} - z_{l+1}^k) + (W_{l+1}^{k+1} - W_{l+1}^k)^T z_{l+1}^k\| + \|(W_{l+1}^{k+1})^T (b_{l+1}^{k+1} - b_{l+1}^k) + (W_{l+1}^{k+1} - W_{l+1}^k)^T b_{l+1}^k\| \\
&+ \|\tau_l^{k+1} \circ (a_l^{k+1} - a_l^k)\| \\
&\le \|(W_{l+1}^{k+1} - W_{l+1}^k)^T W_{l+1}^{k+1} a_l^{k+1}\| + \|(W_{l+1}^k)^T (W_{l+1}^{k+1} - W_{l+1}^k) a_l^{k+1}\| + \|(W_{l+1}^k)^T W_{l+1}^k (a_l^{k+1} - a_l^k)\| \\
&+ \|(W_{l+1}^{k+1})^T (z_{l+1}^{k+1} - z_{l+1}^k)\| + \|(W_{l+1}^{k+1} - W_{l+1}^k)^T z_{l+1}^k\| + \|(W_{l+1}^{k+1})^T (b_{l+1}^{k+1} - b_{l+1}^k)\| + \|(W_{l+1}^{k+1} - W_{l+1}^k)^T b_{l+1}^k\| \\
&+ \|\tau_l^{k+1} \circ (a_l^{k+1} - a_l^k)\| \text{ (triangle inequality)} \\
&\le \|W_{l+1}^{k+1} - W_{l+1}^k\| \|W_{l+1}^{k+1}\| \|a_l^{k+1}\| + \|W_{l+1}^k\| \|W_{l+1}^{k+1} - W_{l+1}^k\| \|a_l^{k+1}\| + \|W_{l+1}^k\| \|W_{l+1}^k\| \|a_l^{k+1} - a_l^k\| \\
&+ \|W_{l+1}^{k+1}\| \|z_{l+1}^{k+1} - z_{l+1}^k\| + \|W_{l+1}^{k+1} - W_{l+1}^k\| \|z_{l+1}^k\| + \|W_{l+1}^{k+1}\| \|b_{l+1}^{k+1} - b_{l+1}^k\| + \|W_{l+1}^{k+1} - W_{l+1}^k\| \|b_{l+1}^k\| \\
&+ \|\tau_l^{k+1}\| \|a_l^{k+1} - a_l^k\| \text{ (Cauchy-Schwarz inequality)}
\end{aligned}$$

and the optimality condition of equation 6 yields

$$0 \in \nabla\phi_{a_l^k}(a_l^k, W_{l+1}^k, z_{l+1}^k, b_{l+1}^k) + \tau_l^{k+1} \circ (a_l^{k+1} - a_l^k) + \partial\mathbb{I}(h_l(z_l^{k+1}) - \varepsilon_l \le a_l^{k+1} \le h_l(z_l^{k+1}) + \varepsilon_l)$$

Because a_l^{k+1},W_{l+1}^k,W_{l+1}^{k+1}, z_{l+1}^k and b_{l+1}^k are bounded, $\|\partial F_{a_l^{k+1}}\|$ can be upper bounded by a linear combination of $\|W_l^{k+1} - W_l^k\|$,$\|b_l^{k+1} - b_l^k\|$,$\|z_l^{k+1} - z_l^k\|$ and $\|a_l^{k+1} - a_l^k\|$.

□