Convergence Rate Analysis of Viscosity Approximation based Gradient Algorithms

Prayas Jain[1], Mridula Verma[2] and K.K Shukla[3]
Indian Institute of Technology (BHU) Varanasi[1,3]
Institute for Development and Research in Banking Technology (IDRBT), Hyderabad[2]
prayas.jain.cse14@iitbhu.ac.in[1], vmridula@idrbt.ac.in[2], kkshukla.cse@iitbhu.ac.in[3]

Abstract

Proximal Algorithms are known to be popular in solving non-smooth convex loss minimization framework due to their low iteration costs and good performance. Convergence rate analysis is an essential part in the process of designing new proximal methods. In this paper, we present a viscosity-approximation-based proximal gradient algorithm and prove its linear convergence rate. We also present its accelerated variant and discuss the condition for the improved convergence rate. These algorithms are applied to solve the problem of multiclass image classification problem. CIFAR10, a popular publicly available benchmark real image classification dataset is used to experimentally validate our theoretical proofs, and the classification performances are compared with that of the state-of-the-art algorithms. To the best of our knowledge, it is the first time that the viscosity-approximation concept is applied to a multiclass classification problem.

I. Introduction

The problem of image/object categorization has been considered to be an important problem in the field of machine learning and computer vision. Among a number of frameworks for this problem, we consider the regularized convex loss minimization framework designed as follows:

$$\min_{x\in\mathbb{R}^d} \quad F(x) = h(x) + \lambda g(x). \tag{1}$$

where d is the dimension of each sample, $h(\cdot) : \mathbb{R}^d \to \mathbb{R}$ is a smooth convex loss function with L-Lipschitz continuous gradient, $g(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ may be a non-smooth convex regularization function and λ is a regularization parameter.

It is assumed that the regularization function $g(\cdot)$ is a simple function, i.e. the exact value of its proximity operator can be computed. A number of first order optimization techniques are available to solve this framework, such as incremental method [1], coordinate descent [2], mirror descent [3], smoothing [4], homotopy [5] and proximal methods [6], [7] to mention a few. In addition, the convergence analysis of such algorithms has also been an important research direction [8], [9], [10], [11]. We concentrate on the proximal methods, where the convergence rate of traditional proximal gradient algorithm (PGA) is $O(1/k)$, where k is the number of iteration. In past decade, acceleration gradient algorithm (AGA) with a number of variants have been proposed with the convergence rate $O(1/k^2)$. Vast applications of AGA for the purpose of classification include [12], [13], [14]. In literature there exists a number of application areas where proximal algorithms are being successfully applied, such as [15], [16], [17], [18].

Various advantages of forward backward algorithms (FBA) motivated researchers to further explore various modifications and generalizations of such methods. Recently, in [19], authors proposed the viscosity-approximation-based proximal gradient algorithm (VGA) as well as its accelerated variant viscosity-approximation-based accelerated gradient algorithm (VAGA) to solve the problem of multitask regression. Although, a number of research has been done in viscosity-approximation-based methods, the analysis of the asymptotic rate of convergence of the viscosity-approximation-based accelerated gradient algorithm is still an open problem. In this paper we not only analyzed the convergence rate of VAGA, we take the experimental results with this algorithm to the next level and present a detail empirical result analysis with the multi-class image classification problem as well.

The contribution of this paper is three-fold:

- We discuss the boundedness of the sequence generated by the new **VGA** and **VAGA** algorithms using an approach different from [19].
- We present detailed proof of convergence rates for both the algorithms.
- Both of these algorithms are applied to the multi-class image classification problem and experimental results with a publicly available benchmark image classification dataset, CIFAR10 are presented.

The organization of the paper is as follows. In the next section, we will discuss the mathematical background of the problem in hand, the related concepts and notations used throughout this paper. In section 3, we will introduce the VGA and the VAGA algorithm for the problem of image classification, and present the boundedness property and convergence rate of the algorithm. The experimental setup and result analysis will be given in section 4. Finally we conclude our work in section 5.

II. Preliminaries

Let $\mathcal{H}$ be a Hilbert space and $T : \mathcal{H} \to \mathcal{H}$ be an operator. T is called an $L-$Lipschitz operator if there exists $L \in [0, \infty)$ such that

$$\|Tx - Ty\| \le L\|x - y\|, \qquad x, y \in \mathcal{H}.$$

An $L-$Lipschitz operator is called a non-expansive operator if $L = 1$ and contraction if $L < 1$. T is monotone if it satisfies,

$$(x, y), (x', y') \in G(T) \Rightarrow \langle x - x', y - y'\rangle \ge 0,$$

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

where $G(T) = \{(x, y) \in \mathcal{H} \times \mathcal{H} : x \in D(T), y \in Tx\}$ is its graph, which is also a monotone set in $\mathcal{H} \times \mathcal{H}$, and $D(T)$ is the domain of operator T. Operator T is *maximal monotone operator* if the graph $G(T)$ is not properly contained in the graph of any other monotone operator. An example of maximal monotone operator is the sub-differential of a convex function. It is well-known that for any $c > 0$, the resolvent J_c^T defined as $J_c^T = (I + cT)^{-1}$ of a maximal monotone operator T is a non-expansive operator. Consider $A : \mathcal{H} \to 2^{\mathcal{H}}$ and $B : \mathcal{C} \to \mathcal{H}$ are two maximal monotone operators, where B satisfies the property $(\mathcal{N})$ on $(0, \gamma_{\mathcal{H},B})$. In order to solve the problem of finding zeros of sum of A and B using forward-backward operator splitting technique, for $r \in (0, \gamma_{\mathcal{H},B})$, the forward-backward operator $J_{c_n}^{A,B}$ is defined by

$$J_{c_n}^{A,B} x = J_{c_n}^{A}(I - c_n B)x, \qquad x \in \mathcal{C}. \tag{2}$$

With respect to problem (1), the forward backward operator $J_{c_n}^{A,B}$ is written as follows:

$$J_{\rho c_n}^{h,g}(x_i) = \text{prox}_{\rho c_n g}(x_i - c_n \nabla h(x_i)), \tag{3}$$

where c_n is a regularization sequence in $(0, \gamma_{\mathcal{H},\nabla g})$. With $x_1 \in \mathbb{R}^d$, the traditional proximal gradient algorithm is defined as the following iterative algorithm,

$$x_{n+1} = J_{\rho c_n}^{h,g}(x_n) \tag{4}$$

In order to accelerate the convergence of traditional proximal gradient algorithm, accelerated proximal gradient algorithm is proposed as follows,

$$\begin{aligned} y_n &= x_n - \beta_n(x_n - x_{n-1}) \\ x_{n+1} &= J_{\rho c_n}^{h,g}(y_n) \end{aligned} \tag{5}$$

where $\beta_n = \frac{t_n - 1}{t_{n+1}}$ with $t_{n+1} = \frac{1+\sqrt{1+4t_n^2}}{2}$ as defined in [6]. In this paper, we present a viscosity-approximation based proximal gradient algorithm and its accelerated variant and discuss their convergence rates.

III. Viscosity-Approximation-based Gradient Algorithms

Let prox_{cg} be the proximity operator w.r.t. g in (1), f be a contraction mapping with contraction factor $\kappa_f \in [0, 1)$, $\{\alpha_n\}$ is a sequence in $(0, 1]$ and c is a value in $(0, 2/L)$. We define $T = \text{prox}_{cg}(I - c\nabla h)$ and $T_n = \text{prox}_{c_n g}(I - c_n \nabla h)$, where $\lim\limits_{n\to\infty} c_n = c$. For any x_0 and $x_1 \in \mathbb{R}^d$, we define iterative schemes for VGA and VAGA [19] for sequence $\{x_n\}$ as follows:

$$\begin{cases} x_{n+1} &= T_n y_n \\ y_n &= (1 - \alpha_n)x_n + \alpha_n f(x_n) \end{cases} \tag{6}$$

$$\begin{cases} x_{n+1} &= T_n y_n \\ y_n &= (1 - \alpha_n)z_n + \alpha_n f(z_n) \\ z_n &= x_n - (x_n - x_{n-1})\frac{t_n - 1}{t_{n+1}} \\ t_{n+1} &= \frac{1+\sqrt{1+4t_n^2}}{2} \end{cases} \tag{7}$$

The viscosity can be adjusted by the parameter κ_f. The pseudo code of the corresponding proximal algorithms are given in algorithms 1 and 2.

Algorithm 1: VGA

```
Data: Data, $\rho$
Result: $x_{n+1}$
begin
    $x_0 = x_1 \in \mathbb{R}^d$, $c_0 = 1, n = 0, \alpha_0 = 0$;
    repeat
        $n = n + 1$;
        Find $c_n$ using backtracking step-size rule and compute $\alpha_n$;
        $y_n = (1 - \alpha_n)x_n + \alpha_n f(x_n)$;
        $x_{n+1} = \text{prox}_{\rho c_n \|\cdot\|_2} y_n$;
    until converge;
```

Algorithm 2: VAGA

```
Data: Data, $\rho$
Result: $x_{n+1}$
begin
    $x_0 = x_1 \in \mathbb{R}^d$, $c_0 = 1$, $t_0 = 0, n = 0, \alpha_0 = 0$;
    repeat
        $n = n + 1$;
        Find $c_n$ using backtracking step-size rule and compute $\alpha_n$;
        $t_{n+1} = \frac{1+\sqrt{1+4t_n^2}}{2}$;
        $z_n = x_n - (x_n - x_{n-1})\frac{t_n - 1}{t_{n+1}}$;
        $y_n = (1 - \alpha_n)z_n + \alpha_n f(z_n)$;
        $x_{n+1} = \text{prox}_{\rho c_n \|\cdot\|_2} y_n$;
    until converge;
```

A. Mathematical analysis

It has been proved that the following result holds in [6], where $L \geq L(h)$, $L(h)$ being the lipschitz continous gradient of the optimization function.

$$F(x) - F(T(y)) \geq \frac{L}{2}\|T(y) - y\|^2 + L\langle y - x, T(y) - y\rangle \tag{8}$$

B. Boundedness Proof of VAGA

Theorem 3.1: Let x_n, y_n, z_n be the sequence generated from VAGA. Then for n>=1, x_n, y_n, z_n are bounded. Boundedness of a sequence is defined as follows: A sequence $\{a_n\}$ is said to be bounded if and only if $\exists M \in \mathbb{R}$ such that

$$\|a_n\| \leq M \quad \forall n \in N$$

Let $\beta_n = \frac{t_n - 1}{t_{n+1}}$. We take t_n as the nesterov acceleration sequence. However, the boundedness is independent of the choice of acceleration sequence, and an alternate sequence might be experimentally chosen for better results. In order to prove the boundedness of the generated sequence, we initiate as follows,

$$\begin{aligned} \|x_{n+1} - x^*\| = \|T_n y_n - x^*\| &\leq \|y_n - x^*\| \\ &\leq (1 - \alpha_n)\|z_n - x^*\| + \alpha_n\|f(z_n) - x^*\| \\ &\leq (1 - (1 - \kappa_f)\alpha_n)\|z_n - x^*\| + \|f(x^*) - x^*\| \\ &\leq max\{\|z_n - x^*\|, \frac{\|fx^* - x^*\|}{1 - \kappa_f}\} \end{aligned}$$

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

Now, from 7, we can write,

$$\begin{aligned}\|z_n - x^*\| &= \|x_n + \frac{t_n - 1}{t_{n+1}}(x_n - x_{n-1}) - x^*\| \\ &= \|(1+\beta_n)x_n - \beta_n x_{n-1} - x^*\| \\ &\le (1+\beta_n)\|x_n - x^*\| + \beta_n\|x_{n-1} - x^*\| \\ &\le (1+2\beta_n)max\{\|x_n - x^*\|, \|x_{n-1} - x^*\|\}\end{aligned}$$

Let the upper bound of our momentum sequence $\{\beta_n\}$ (a hyperparameter) be μ_1. *Note:* In most of the latest implementations, like deep learning models or other computational frameworks, it has been observed that momentum is generally kept constant, closer to 1 (around 0.9). Keeping it around 0.9 works well and almost no improvement is observed in shifting it. Hence, models use a constant sequence as the momentum sequence, where $\beta_n = \mu_1$. Let μ_2 be another constant. Then by replacing $1 + 2\beta_n \le \mu_2$, we get,

$$\|z_{n+1} - x^*\| \le max\{\mu_2\|x_n - x^*\|, \mu_2\|x_{n-1} - x^*\|\}$$

$$\begin{aligned}\|x_{n+1} - x^*\| \le max\{\mu_2\|x_n - x^*\|, \mu_2\|x_{n-1} - x^*\|, \\ \frac{\|fx^* - x^*\|}{1 - \kappa_f}\}\end{aligned} \tag{9}$$

Extrapolating, and replacing, we will get

$$\|x_1 - x^*\| \le max\{\mu_2\|x_0 - x^*\|, \mu_2\|x_{-1} - x^*\|, \frac{\|fx^* - x^*\|}{1 - \kappa_f}$$

Since, during initialization $x_0 = x_{-1} =$ Initialization

$$\|x_1 - x^*\| \le max\{\mu_2\|x_0 - x^*\|, \frac{\|fx^* - x^*\|}{1 - \kappa_f}\}$$

And similarly,

$$\|x_2 - x^*\| \le max\{\mu_2\|x_0 - x^*\|, \frac{\|fx^* - x^*\|}{1 - \kappa_f}\}$$

Hence,

$$\|x_{n+1} - x^*\| \le max\{\mu_2\|x_0 - x^*\|, \frac{\|fx^* - x^*\|}{1 - \kappa_f}\}$$

Thus, $\{x_n\}$ and similarly $\{y_n\}$, $\{z_n\}$ are bounded.

IV. Proof Convergence Rate VGA

Theorem 4.1: Let x_n, y_n be the sequence generated from VGA. Then for n>=1,

$$F(x_n) - F(x^*) \le \frac{C}{n}$$

where $C \ge 0$, $\forall x^* \in X_*$

Proof: Consider $\beta L(f) < L_n < \psi L(f) \forall n$. Substitute, $x = x^*$, $y = y_n$ and $L = L_{n+1}$ in (8), using pythagoras theorem

$$\begin{aligned}\frac{2}{\psi L(f)}(F(x^*) - F(x_{n+1})) \ge \|x_{n+1} - x^*\|^2 - \|y_n - x^*\|^2 \\ \ge (\|x_{n+1} - x^*\|^2 - \|x_n - x^*\|^2) \\ + \alpha_n(\|x_n - fx_n\|^2 + 2\langle x_n - fx_n, fx_n - x^*\rangle)\end{aligned} \tag{10}$$

Substitute $x = x_n$, $y = y_n$ and $L = L_{n+1}$ in 8, we get

$$\begin{aligned}\frac{2}{\beta L(f)}(F(x_n) - F(x_{n+1})) &\ge \|x_{n+1} - x_n\|^2 - \|y_n - x_n\|^2 \\ &\ge \|x_{n+1} - x_n\|^2 - \alpha_n\|fx_n - x_n\|^2 \\ &\ge \|x_{n+1} - fx_n\|^2 + 2\langle x_{n+1} - fx_n, fx_n - x_n\rangle\end{aligned}$$

Case 1: RHS is positive, i.e., $\|x_{n+1} - fx_n\|^2 + 2\langle x_{n+1} - fx_n, fx_n - x_n\rangle \ge 0$. Then,

$$\begin{aligned}\frac{2}{\beta L(f)}(nF(x_n) - (n+1)F(x_{n+1}) + F(x_{n+1})) \\ \ge n(\|x_{n+1} - fx_n\|^2 + 2\langle x_{n+1} - fx_n, fx_n - x_n\rangle)\end{aligned}$$

Summing over $n = 0 \cdots k-1$,

$$\begin{aligned}\frac{2}{\beta L(f)}(-kF(x_k) + \sum_{n=0}^{k-1} F(x_{n+1})) \\ \ge \sum_{n=0}^{k-1} n(\|x_{n+1} - fx_n\|^2 + 2\langle x_{n+1} - fx_n, fx_n - x_n\rangle)\end{aligned} \tag{11}$$

Multiplying both sides by $\frac{\beta}{\psi}$,

$$\begin{aligned}\frac{2}{\psi L(f)}(-kF(x_k) + \sum_{n=0}^{k-1} F(x_{n+1}) \\ \ge \frac{\beta}{\psi}\sum_{n=0}^{k-1} n(\|x_{n+1} - fx_n\|^2 + 2\langle x_{n+1} - fx_n, fx_n - x_n\rangle)\end{aligned} \tag{12}$$

Similarly, from (10),

$$\begin{aligned}\frac{2}{\psi L(f)}(F(x^*) - F(x_{n+1})) \\ \ge (\|x_{n+1} - x^*\|^2 - \|x_n - x^*\|^2) \\ + \alpha_n(\|x_n - fx_n\|^2 + 2\langle x_n - fx_n, fx_n - x^*\rangle)\end{aligned} \tag{13}$$

Summing over $n = 0, \cdots k-1$,

$$\begin{aligned}\frac{2}{\psi L(f)}(kF(x^*) - \sum_{n=0}^{k-1} F(x_{n+1})) \\ \ge \|x_k - x^*\|^2 - \|x_0 - x^*\|^2 + \sum_{n=0}^{k-1} \alpha_n(\|x_n - fx_n\|^2 \\ + 2\langle x_n - fx_n, fx_n - x^*\rangle)\end{aligned} \tag{14}$$

Also, since the RHS is -ve, we can write,

$$\begin{aligned}\|x_0 - x^*\|^2 \ge \|x_k - x^*\|^2 + \sum_{n=0}^{k-1} \alpha_n(\|x_n - fx_n\|^2 + \\ 2\langle x_n - fx_n, fx_n - x^*\rangle)\end{aligned} \tag{15}$$

Adding 14 and 12, we get,

$$\begin{aligned}\frac{2}{\psi L(f)}(kF(x^*) - kF(x_k)) \ge \frac{\beta}{\psi}\sum_{n=0}^{k-1} n(\|x_{n+1} - fx_n\|^2 + \\ 2\langle x_{n+1} - fx_n, fx_n - x_n\rangle) + \|x_k - x^*\|^2 - \|x_0 - x^*\|^2 \\ + \sum_{n=0}^{k-1} \alpha_n(\|x_n - fx_n\|^2 + 2\langle x_n - fx_n, fx_n - x^*\rangle)\end{aligned}$$

$$\frac{2k}{\psi L(f)}(F(x_k) - F(x^*)) \leq -\frac{\beta}{} \sum_{n=0}^{k-1} n(\|x_{n+1} - fx_n\|^2 + \tag{16}$$

$$2\langle x_{n+1} - fx_n, fx_n - x_n\rangle) - \|x_k - x^*\|^2$$

$$+ \|x_0 - x^*\|^2 - \sum_{n=0}^{k-1} \alpha_n(\|x_n - fx_n\|^2 +$$

$$2\langle x_n - fx_n, fx_n - x^*\rangle)$$

$$\leq \|x_0 - x^*\|^2 - \|x_k - x^*\|^2 - \sum_{n=0}^{k-1} \alpha_n(\|x_n - fx_n\|^2) \tag{17}$$

From 16 and 15, we can write,

$$\frac{2}{\psi L(f)}(kF(x^*) - kF(x_k)) \leq C \tag{18}$$

where $C = \|x_0 - x^*\|^2 - \|x_k - x^*\|^2 - \sum_{n=0}^{k-1} \alpha_n(\|x_n - fx_n\|^2 + 2\langle x_n - fx_n, fx_n - x^*\rangle) \geq 0$.
Also, since x_k converges to x^*, loss function F will trivially satisfy, $F(x_k) - F(x^*) >= 0$. That is, $C >= 0$ Thus,

$$F(x_k) - F(x^*) \leq \frac{C\psi L(f)}{2k} \tag{19}$$

Case 2: RHS is negative, i.e., $\|x_{n+1} - fx_n\|^2 + 2\langle x_{n+1} - fx_n, fx_n - x_n\rangle \leq 0$. Since LHS is +ve,

$$\frac{2}{\beta L(f)}(F(x_n) - F(x_{n+1})) \geq 0$$

Summing over $n = 0, \cdots k - 1$,

$$\frac{2}{\beta L(f)}(-kF(x_k) + \sum_{n=0}^{k-1} F(x_{n+1}) \geq 0 \tag{20}$$

From 20 and 14,

$$\frac{2k}{\psi L(f)}(F(x_k) - F(x^*)) \leq \|x_0 - x^*\|^2 - \|x_k - x^*\|^2 - \sum_{n=0}^{k-1} \alpha_n \|x_n - fx_n\|^2 \tag{21}$$

From 15 and 21, we again get,

$$\frac{2}{\psi L(f)}(kF(x^*) - kF(x_k)) \leq C$$

Case 3: RHS assumes both positive and negative values in different iteration steps
In this case, the proof given for cases 1 and 2 can be combined to get the same result.

V. Proof Convergence Rate VAGA

Theorem 5.1: Let x_n, y_n, z_n be the sequence generated from VAGA. Then for n>=1,

$$F(x_n) - F(x^*) \leq \frac{2\alpha L(h)\|x_0 - x^*\|^2}{(k+1)^2}, \forall x^* \in X_*$$

if

$$\begin{bmatrix} t_{n+1}^2 \\ t_n^2 \\ t_n t_{n+1} \\ t_{n+1} \\ t_n \\ 1 \end{bmatrix}^T \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \\ -2 & 2 & 0 & -1 & 1 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 & 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 & 0 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} \langle x_n, y_n\rangle \\ \|x_n\|^2 \\ \|y_n\|^2 \\ \langle x_n, x^*\rangle \\ \langle x_{n-1}, y_n\rangle \\ \langle x_n, x_{n-1}\rangle \\ \langle x_{n-1}, x^*\rangle \\ \langle y_n, x^*\rangle \end{bmatrix} \geq 0 \tag{22}$$

Proof: let $v_n = F(x_n) - F(x^*)$. Substituting $x = x^*$, $y = y_n$ in 8 , we get

$$-2L_{n+1}^{-1} v_{n+1} \geq \|x_{n+1} - x^*\|^2 - \|y_n - x^*\|^2$$

Using the expression $\|b - a\|^2 + \langle b - a, a - c\rangle = \|b - c\|^2 - \|a - c\|^2$,

$$-2L_{n+1}^{-1} v_{n+1} \geq \|x_{n+1} - y_n\|^2 + 2\langle x_{n+1} - y_n, y_n - x^*\rangle$$

Similarly, substituting $x = x_n$, $y = y_n$ in 8, we get,

$$\frac{2}{\beta L(f)}(F(x_n) - F(x_{n+1})) \geq \|x_{n+1} - x_n\|^2 - \|y_n - x_n\|^2 \tag{23}$$

$$2L_{n+1}^{-1}(v_n - v_{n+1}) \geq \|x_{n+1} - y_n\|^2 + 2\langle x_{n+1} - y_n, y_n - x_n\rangle \tag{24}$$

Multiplying (24) by $(t_{n+1} - 1)$ and adding to (10), we get,

$$\frac{2}{L_{k+1}}((t_{n+1} - 1)v_n - t_{n+1}v_{n+1}) \geq t_{n+1}\|x_{n+1} - y_n\|^2 + 2\langle x_{n+1} - y_n, t_{n+1}y_n - x^* - (t_{n+1} - 1)x_n\rangle$$

$$\frac{2}{L_{k+1}}(t_n^2 v_n - t_{n+1}^2 v_{n+1}) \geq \|t_{n+1}x_{n+1} - y_n\|^2 + \tag{25}$$

$$2t_{n+1}\langle x_{n+1} - y_n, t_{n+1}y_n - x^* - (t_{n+1} - 1)x_n\rangle$$

$$\geq \|t_{n+1}x_{n+1} - (t_{n+1} - 1)x_n - x^*\|^2 - \tag{26}$$

$$\|t_{n+1}y_n - (t_{n+1} - 1)x_n - x^*\rangle$$

let $u_n = t_n x_n - (t_n - 1)x_{n-1} - x^*$. Substituting in (25) we get,

$$\frac{2}{L_{k+1}}(t_n^2 v_n - t_{n+1}^2 v_{n+1}) \geq \|u_{n+1}\|^2$$

$$-\|t_{n+1}y_n - (t_{n+1} - 1)x_n - x^*\|^2$$

$$\geq \|u_{n+1}\|^2 - \|u_n\|^2 + \|u_n\|^2 -$$

$$\|t_{n+1}y_n - (t_{n+1} - 1)x_n - x^*\|^2$$

In order to meet the requirement, we need to prove that the LHS is greater than the quantity $\|u_{n+1}\|^2 - \|u_n\|^2$, for which we need to prove that the quantity $\|u_n\|^2 - \|t_{n+1}y_n - (t_{n+1} - 1)x_n - x^*\|^2$ is positive. Or in other words,

$$\|t_n x_n - (t_n - 1)x_{n-1} - x^*\|^2 -$$

$$\|t_{n+1}y_n - (t_{n+1} - 1)x_n - x^*\|^2 \geq 0$$

Again using the expression $\|b - a\|^2 + \langle b - a, a - c\rangle = \|b - c\|^2 - \|a - c\|^2$, we get,

$$\|t_n x_n - (t_n - 1)x_{n-1} - t_{n+1}y_n - (t_{n+1} - 1)x_n\|^2$$

$$+ 2\langle t_n x_n - (t_n - 1)x_{n-1} - t_{n+1}y_n - (t_{n+1} - 1)x_n, \tag{27}$$

$$t_{n+1}y_n - (t_{n+1} - 1)x_n - x^*\rangle \geq 0$$

TABLE I
COMPARISON OF ACCURACY AND COMPUTATIONAL TIME OF DISCUSSED ALGORITHMS

Optimization	Accuracy			Avg. Computational Time (sec)	
Algorithm	Training	Validation	Testing	Total	Per Epoch
SGA	0.3737	0.3298	0.3572	162.49	10.83
PGA	**0.4110**	0.3660	0.3667	161.96	10.79
VGA	0.3922	0.3764	0.3845	164.71	10.98
FISTA	0.4102	0.3693	0.3802	166.37	11.09
VAGA	0.4104	**0.3895**	**0.3989**	168.80	11.25

Since the first term will be always positive, we now need to prove,

$$\begin{aligned}&\langle (t_{n+1} + t_n - 1)x_n - (t_n - 1)x_{n-1} - t_{n+1}y_n,\\ &\quad t_{n+1}y_n - (t_{n+1} - 1)x_n - x^* \rangle \geq 0\end{aligned}$$

Solving the above equation, we get the final condition for the $O(1/k^2)$ for algorithm as given in equation (22).

VI. EXPERIMENTAL SETUP AND RESULT ANALYSIS

The Viscosity based Accelerated Gradient Algorithm (VAGA) was compared with three other proximal gradient based algorithms and one non-proximal based algorithm (Stochastic Gradient Descent) on CIFAR-10 dataset. We also experimentally verify the proposed condition of convergence for VAGA. We describe the dataset and our model in the next sub-section followed by the results.

A. Dataset Description

The CIFAR-10 dataset [20] consists of 60,000 RGB colored 32×32 pixel images equally distributed into 10 classes. We use 40,000 examples for training, 10,000 for validation and 10,000 for testing the model. We use raw pixel information as features for the model. The pixel value can range from 0-255 for each color. As a pre-processing step, the mean image is subtracted from the dataset, and a bias feature of value 1 is added. Hence, we have a total of 3073 (32×32×3 + 1) input features for the model.

B. Model Description

The model used to show experimental results can be considered as the baseline model for the problem of Image Classification. We use a multi-class logistic regression framework as our model. The framework uses Softmax function as the activation function. The number of classes for the CIFAR-10 dataset are 10. The model returns a score (between 0 and 1) for each class. We mark the class with the highest score as the predicted class, and use it to calculate the accuracy of the model. We optimize the model on cross entropy loss and ℓ_2 regularizer. The reason for using ℓ_2 regularizer is the fact that Stochastic Gradient Descent requires smooth optimization functions, and ℓ_1 regularizer adds non-smoothness to the optimization function. Further, we also observe similar results for ℓ_1 regularizer for proximal algorithms. The cross-entropy loss for the i^{th} example, having y_i class has the form:

$$L_i = -\log\left(\frac{e^{f_i}}{\sum_j e^{f_j}}\right)$$

or equivalently

$$L_i = -f_{y_i} + log \sum_j e^{f_j}$$

where $f_j(z)$ is the softmax function defined as

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}}$$

We train the model in a stochastic fashion, with a batch size of 250 images for 15 epochs. We also experimented with higher epochs, which resulted in no significant increase in accuracy or decrease in loss for any of the discussed algorithms, hence sticking with 15 epochs for comparison. The number of iterations per epoch are equal to the total size of the training dataset divided by the batch size of each iteration, 160 in our experiments. Hence we train for 2400 iterations for each algorithm. The models were trained on Intel(R) Xeon(R) E5-2420 v2 CPU with a 2.20 GHz clock speed. The images for each batch are chosen randomly, using a uniform distribution from the set of training images. Hyper-parameter optimization is done using random search. After reducing the optimal parameters for each algorithm, we run the experiments 30 times and show the training and validation accuracy statistics using box-plots shown below. [1]

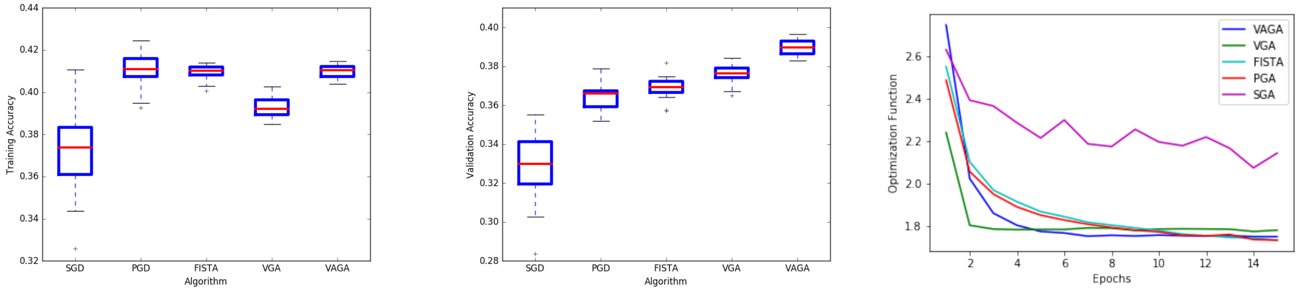
The deduced condition for convergence is checked in each iteration for VAGA and is found to be satisfied for **99.75%** of the total iterations. The optimization algorithms may be applied and compared on complex deep learning systems to challenge the state of the art. We leave that as future work, and focus on establishing the competitiveness of our optimization algorithm.

The detailed results are presented in table I. We report median training and validation accuracies along with the final testing accuracy. We observe better results show by VAGA optimization scheme compared to other techniques by a slight margin. Simple stochastic gradient descent show lesser validation and testing accuracy compared to other techniques. Both VGA and VAGA perform the best, highlighting the usefulness of the viscosity based component in the optimization techniques.

We also report the average computational time elapsed by the iterations in terms of the total and per epoch time taken by the different algorithms. While SGD has least total and

[1] To ensure our experiments and results are reproducible, we will be releasing the Python Code of our experiments along with all the fine-tuned hyperparameters.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE



(a) Box Plots for Training Accuracy

(b) Box Plots for Validation Accuracy

(c) Convergence of Optimization Function

Fig. 1. Accuracy and Convergence Results for the CIFAR10 dataset. 1(a) and 1(b), Experiments repeated 30 times for generation of BoxPlots. 1(c) The optimization function used is cross entropy loss combined with l2 regularizer.

per epoch time, due to its relative simplicity, the time for the proposed technique is also comparable, as shown in I. The higher per epoch cost arises due to extra computation while calculating the acceleration step. We also report the graph for the convergence of algorithms. It should be noted that since the algorithms are used in a stochastic fashion, and the training is done in batches due to complexity issues, the training loss might increase for some iterations, but the general trend is of convergence. VAGA and VGA algorithm convergence faster than other algorithms, and all the proximal algorithms converge much faster than standard SGA.

Due to the faster convergence of VAGA algorithm, the overall computational cost of training a model might be lower than using other non-accelerated variations due to lesser number of epochs despite having a higher per epoch cost. We leave experimentation with bigger data-sets for future work.

VII. Conclusion

In this paper, we present a viscosity-approximation based gradient and accelerated gradient algorithm to solve the multi-class image classification problem and discuss the convergence rates of both the algorithms. We performed experiments on the real benchmark CIFAR10 dataset. We conclude that the theoretical convergence rates of VGA and VAGA is similar as that of PGA and AGA, however the practical performance of these algorithms are better than the state-of-the-art algorithms in terms of classification accuracy. In future, we aim to apply these algorithms to the popular deep learning frameworks.

References

[1] D. P. Bertsekas, "Incremental proximal methods for large scale convex optimization," *Mathematical programming*, vol. 129, no. 2, p. 163, 2011.

[2] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.

[3] A. Beck and M. Teboulle, "Mirror descent and nonlinear projected subgradient methods for convex optimization," *Operations Research Letters*, vol. 31, no. 3, pp. 167–175, 2003.

[4] Y. Nesterov, "Smooth minimization of non-smooth functions," *Mathematical programming*, vol. 103, no. 1, pp. 127–152, 2005.

[5] D. L. Donoho and Y. Tsaig, "Fast solution of ell_1-norm minimization problems when the solution may be sparse," *IEEE Transactions on Information Theory*, vol. 54, no. 11, pp. 4789–4812, 2008.

[6] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009. [Online]. Available: http://dx.doi.org/10.1137/080716542

[7] A. Chambolle and C. Dossal, "On the convergence of the iterates of the "fast iterative shrinkage/thresholding algorithm"," *Journal of Optimization Theory and Applications*, vol. 166, no. 3, pp. 968–982, 2015. [Online]. Available: http://dx.doi.org/10.1007/s10957-015-0746-4

[8] D. Yang and Y. Liu, "L1/2 regularization learning for smoothing interval neural networks: Algorithms and convergence analysis," *Neurocomputing*, vol. 272, pp. 122–129, 2018.

[9] Q. Meng, W. Chen, Y. Wang, Z.-M. Ma, and T.-Y. Liu, "Convergence analysis of distributed stochastic gradient descent with shuffling," *Neurocomputing*, vol. 337, pp. 46 – 57, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231219300578

[10] M. Verma, P. Jain, and K. K. Shukla, "A new faster first order iterative scheme for sparsity-based multitask learning," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016, pp. 001 603–001 608.

[11] M. Verma and K. Shukla, "Convergence analysis of accelerated proximal extra-gradient method with applications," *Neurocomputing*, vol. 388, pp. 288 – 300, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231220301004

[12] S. Bubeck, Y. T. Lee, and M. Singh, "A geometric alternative to nesterov's accelerated gradient descent," *arXiv preprint arXiv:1506.08187*, 2015.

[13] A. Fawzi, M. Davies, and P. Frossard, "Dictionary learning for fast classification based on soft-thresholding," *International Journal of Computer Vision*, vol. 114, no. 2-3, pp. 306–321, 2015.

[14] N. Ito, A. Takeda, and K.-C. Toh, "A unified formulation and fast accelerated proximal gradient method for classification," *Journal of Machine Learning Research*, vol. 18, no. 16, pp. 1–49, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-274.html

[15] Y. Zhang, G. Zhou, Q. Zhao, A. Cichocki, and X. Wang, "Fast nonnegative tensor factorization based on accelerated proximal gradient and low-rank approximation," *Neurocomputing*, vol. 198, pp. 148–154, 2016.

[16] S. F. Mahmood, M. H. Marhaban, F. Z. Rokhani, K. Samsudin, and O. A. Arigbabu, "Fasta-elm: a fast adaptive shrinkage/thresholding algorithm for extreme learning machine and its application to gender recognition," *Neurocomputing*, vol. 219, pp. 312–322, 2017.

[17] B. Xu and Q. Liu, "Iterative projection based sparse reconstruction for face recognition," *Neurocomputing*, vol. 284, pp. 99–106, 2018.

[18] T. Lin, L. Qiao, T. Zhang, J. Feng, and B. Zhang, "Stochastic primal-dual proximal extragradient descent for compositely regularized optimization," *Neurocomputing*, vol. 273, pp. 516–525, 2018.

[19] M. Verma, D. R. Sahu, and K. K. Shukla, "Vaga: a novel viscosity-based accelerated gradient algorithm," *Applied Intelligence*, Dec 2017. [Online]. Available: https://doi.org/10.1007/s10489-017-1110-1

[20] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.