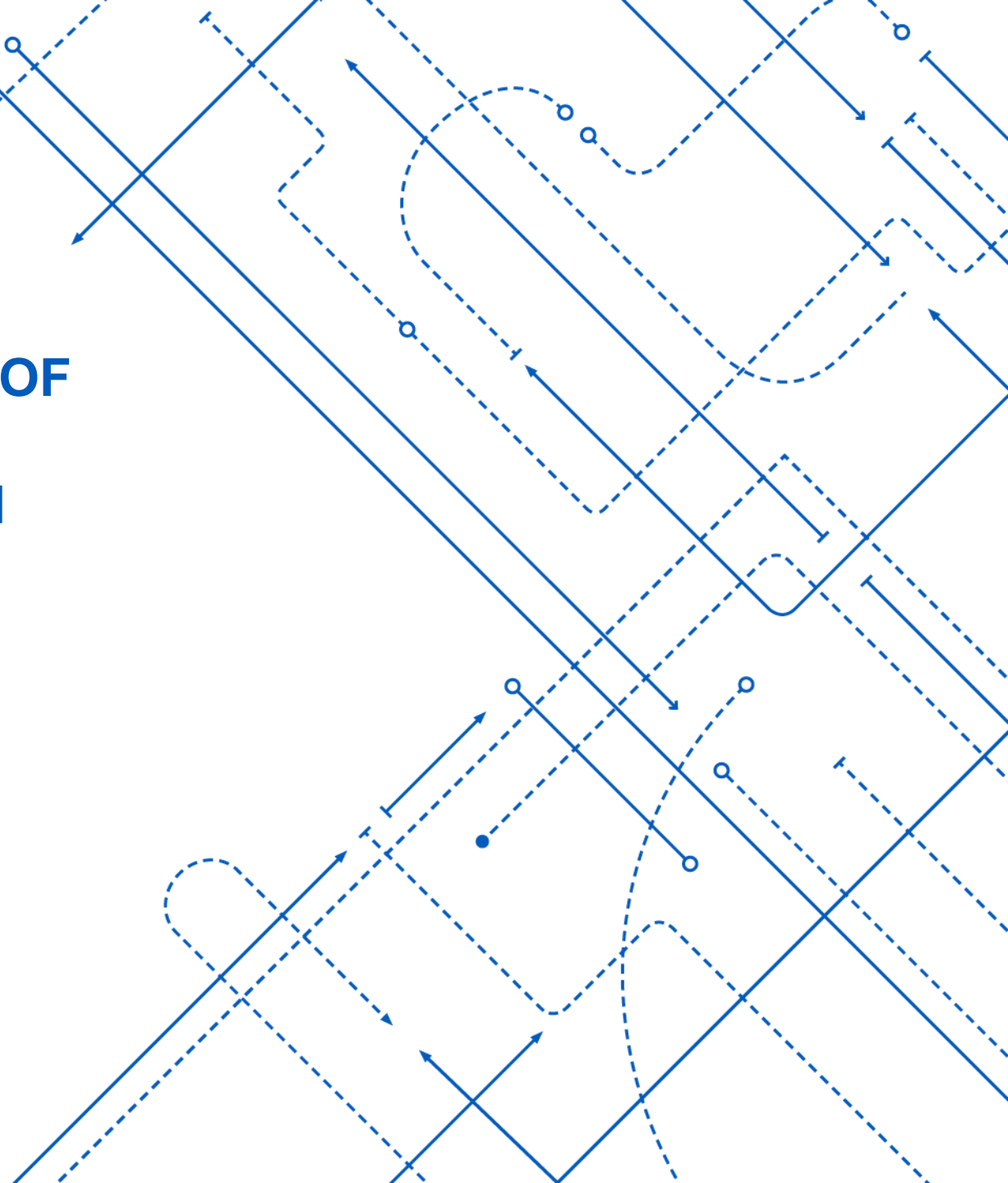
TOWARDS BETTER UNDERSTANDING OF ADAPTIVE GRADIENT ALGORITHMS IN GENERATIVE ADVERSARIAL NETS

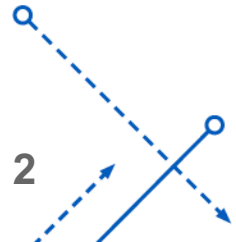
Ashray A Sripathi

ashraysr@buffalo.edu



University at Buffalo The State University of New York

Contents



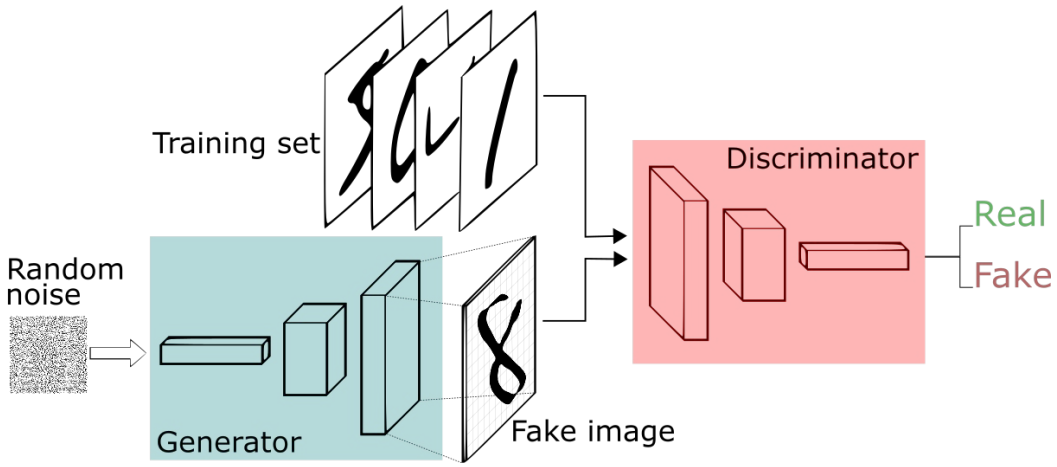
Introduction

- Adaptive gradient algorithms are very popular in training deep neural networks due to their computational efficiency and minimal need for hyper-parameter tuning.
- However, there is not enough evidence showing that adaptive gradient methods converge faster in supervised deep learning tasks.
- Perform worse than SGD on image classification tasks.

Introduction

- GANs are a popular class of generative models.
- They consist of a generator and a discriminator, both of which are defined by deep neural networks.
- They are trained under adversarial cost.
- Adam is the preferred optimizer for training GANs
- GAN is a min-max optimization problem in nature.

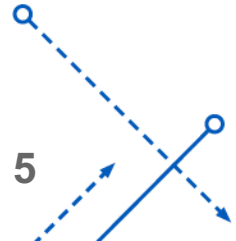
$$\min_{\mathbf{u}\in\mathcal{U}} \max_{\mathbf{v}\in\mathcal{V}} F(\mathbf{u},\mathbf{v}) := \mathbb{E}_{\xi\sim\mathcal{D}}\left[f(\mathbf{u},\mathbf{v};\xi)\right]$$



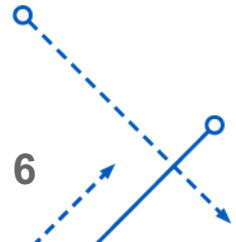
Introduction

- Why do adaptive gradient methods outperform their non-adaptive counterparts in GAN training?
  - Analyze a variant of Optimistic Stochastic Gradient.
  - Proposal of an adaptive variant of Adagrad named Optimistic Adagrad (OAdagrad).
  - Observed that the cumulative stochastic gradient grows at a slow rate.

- **The main goal is to design stochastic first-order algorithms with low iteration complexity, low per-iteration cost and suitable for a general class of non-convex non-concave min-max problems.**

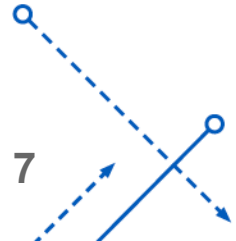


Contents



Related Work

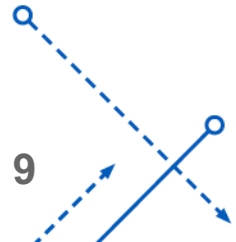
- Dang & Lan, 2015 designed a first-order deterministic algorithm with a non-asymptotic guarantee.
- Iusem et al., 2017 develops a stochastic extra-gradient algorithm that enjoys $O(\epsilon-4)$ iteration complexity.
  - These are extra gradient methods, causing increased computational expense for GANs
- Daskalakis et al., 2017 considers both min max settings and GAN training.
  - Algorithm shown only for bilinear class of problems.



	Assumption	Setting	IC	PC	Guarantee
Extragradient (Iusem et al., 2017)	pseudo-monotonicity [3]	stochastic	$O(\epsilon^{-4})$	$2T_g$	ϵ-SP
OMD (Daskalakis et al., 2017)	bilinear	deterministic	N/A	T_g	asymptotic
AvgPastExtraSGD (Gidel et al., 2018)	monotonicity	stochastic	$O(\epsilon^{-2})$	T_g	ϵ-DG
OMD (Mertikopoulos et al., 2018)	coherence	stochastic	N/A	$2T_g$	asymptotic
IPP (Lin et al., 2018)	MVI has solution	stochastic	$O(\epsilon^{-6})$	T_g	ϵ-SP
Alternating Gradient (Gidel et al., 2019)	bilinear form [4]	deterministic	$O(\log(1/\epsilon)$	T_g	ϵ-optim
SVRE (Chavdarova et al., 2019)	strong-monotonicity finite sum	stochastic finite sum	$O(\log(1/\epsilon)$	$(n+\frac{L}{\mu})T_g$ [5]	ϵ-optim
Extragradient (Azizian et al., 2019)	strong-monotonicity	deterministic	$O(\log(1/\epsilon)$	$2T_g$	ϵ-optim
OSG (this work)	MVI has solution	stochastic	$O(\epsilon^{-4})$	T_g	ϵ-SP
OAdagrad (this work)	MVI has solution	stochastic	$O(\epsilon^{-\frac{2}{1-\alpha}})$	T_g	ϵ-SP

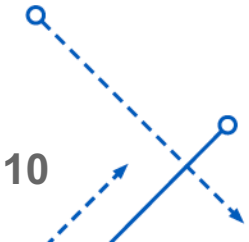
Table 1: Summary of different algorithms with IC (Iteration Complexity), PC (Per-iteration Complexity) to find ϵ-SP (ϵ-first-order Stationary Point), ϵ-DG (ϵ-Duality Gap, i.e. a point $(\hat{\mathbf{u}}, \hat{\mathbf{v}})$ such that $\max_{\mathbf{v}} F(\hat{\mathbf{u}}, \mathbf{v}) - \min_{\mathbf{u}} F(\mathbf{u}, \hat{\mathbf{v}}) \leq \epsilon$), or ϵ-optim (ϵ-close to the set of optimal solution). T_g stands for the time complexity for invoking one stochastic first-order oracle.

Contents



Preliminaries and Notations

- The main tool used in our analysis is variational inequality.
- Stampacchia Variational Inequality (SVI)
  - finding $y \in K$ such that $\langle F(y), x - y\rangle \geq 0$, for all $x \in K$
  - SVI expresses a local condition for function F at one point.
- Minty Variational Inequality(MVI)
  - finding $y \in K$ such that $\langle F(x), y - x\rangle$, for all $x \in K$.
  - In MVI, we must deal with all function values taken at any $x \in K$.



Preliminaries and Notations

Definition 1 (Monotonicity). *An operator T is monotone if $\langle T(\mathbf{x})-T(\mathbf{y}), \mathbf{x}-\mathbf{y}\rangle \geq 0$ for $\forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$. An operator T is pseudo-monotone if $\langle T(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle \geq 0 \Rightarrow \langle T(\mathbf{y}), \mathbf{y}-\mathbf{x}\rangle \geq 0$ for $\forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$. An operator T is γ-strongly-monotone if $\langle T(\mathbf{x})-T(\mathbf{y}), \mathbf{x}-\mathbf{y}\rangle \geq \frac{\gamma}{2}\|\mathbf{x}-\mathbf{y}\|^2$ for $\forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$.*

- Assuming SVI has a solution and pseudo-monotonicity of the operator T we can imply that MVI(T, X) has a solution.
- Solving SVI(T, X) is NP-hard in general and hence we resort to finding an ϵ-first-order stationary point.

Definition 2 (ϵ-First-Order Stationary Point). *A point $\mathbf{x} \in \mathcal{X}$ is called ϵ-first-order stationary point if $\|T(\mathbf{x})\| \leq \epsilon$.*

Preliminaries and Notations

Assumption 1. *(i). T is L-Lipschitz continuous, i.e.* $\|T(\mathbf{x}_1) - T(\mathbf{x}_2)\|_2 \leq L\|\mathbf{x}_1 - \mathbf{x}_2\|_2$ *for* $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$.

(ii). MVI$(T, \mathcal{X})$ has a solution, i.e. there exists $\mathbf{x}_*$ *such that* $\langle T(\mathbf{x}), \mathbf{x} - \mathbf{x}_* \rangle \geq 0$ *for* $\forall \mathbf{x} \in \mathcal{X}$.

(iii). For $\forall \mathbf{x} \in \mathcal{X}$, $\mathbb{E}\left[T(\mathbf{x}; \xi)\right] = T(\mathbf{x})$, $\mathbb{E}\left\|T(\mathbf{x}; \xi) - T(\mathbf{x})\right\|^2 \leq \sigma^2$.

- Assumptions (i) and (iii) are commonly used assumptions in the literature of variational inequalities and non-convex optimization.
- For nonconvex minimization problem, it has been shown that assumption (ii) holds while using SGD to learn neural networks

Contents

OPTIMISTIC STOCHASTIC GRADIENT

Algorithm 1 Optimistic Stochastic Gradient (OSG)

Input: $\mathbf{z}_0 = \mathbf{x}_0 = 0$
for $k = 1, \ldots, N$ **do**
  $\mathbf{z}_k = \Pi_{\mathcal{X}} \left[\mathbf{x}_{k-1} - \eta \cdot \frac{1}{m_{k-1}} \sum_{i=1}^{m_{k-1}} T(\mathbf{z}_{k-1}; \xi_{k-1}^i) \right]$
  $\mathbf{x}_k = \Pi_{\mathcal{X}} \left[\mathbf{x}_{k-1} - \eta \cdot \frac{1}{m_k} \sum_{i=1}^{m_k} T(\mathbf{z}_k; \xi_k^i) \right]$
end for

- In stochastic extra gradient method, the extra call on z^k allows the algorithm to "anticipate" the landscape of T.
- In contrast, $\{x^k\}$ is an ancillary sequence in OSG and the stochastic gradient is only computed over the sequence of $\{z^k\}$.

Theorem 1. *Suppose that Assumption 1 holds. Let* $r_\alpha(\mathbf{z}_k) = \|\mathbf{z}_k - \Pi_{\mathcal{X}}\left(\mathbf{z}_k - \alpha T(\mathbf{z}_k)\right)\|$. *Let* $\eta \leq 1/9L$ *and run Algorithm 1 for* N *iterations. Then we have*

$$\frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\left[r_\eta^2(\mathbf{z}_k)\right] \leq \frac{8\|\mathbf{x}_0 - \mathbf{x}_*\|^2}{N} + \frac{100\eta^2}{N}\sum_{k=0}^{N}\frac{\sigma^2}{m_k},$$

- (Increasing Minibatch Size) Let $\eta = \frac{1}{9L}$, $m_k = k + 1$. To guarantee $\frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\|T(\mathbf{z}_k)\|_2^2 \leq \epsilon^2$, the total number of iterations is $N = \widetilde{O}(\epsilon^{-2})$, and the total complexity is $\sum_{k=1}^{N} m_k = \widetilde{O}(\epsilon^{-4})$, where $\widetilde{O}(\cdot)$ hides a logarithmic factor of ϵ.
- (Constant Minibatch Size) Let $\eta = \frac{1}{9L}$, $m_k = 1/\epsilon^2$. To guarantee $\frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\|T(\mathbf{z}_k)\|_2^2 \leq \epsilon^2$, the total number of iterations is $N = O(\epsilon^{-2})$, and the total complexity is $\sum_{k=0}^{N} m_k = O(\epsilon^{-4})$.

Contents

Optimistic Adagrad(OAdagrad)

- Adagrad For Minimization Problems
  - The main objective in Adagrad is to solve the following minimization problem:

$$\min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}) = \mathbb{E}_{\zeta\sim\mathcal{P}} f(\mathbf{w};\zeta)$$

  - Where the weight update rule is given by

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta H_t^{-1}\hat{\mathbf{g}}_t,$$

- Adagrad dynamically incorporates knowledge of history gradients to perform more informative gradient-based learning.
- Cannot be directly applied to solve non-convex non-concave minmax problems with provable guarantee.

Optimistic Adagrad

- Optimistic Adagrad For Min-max Optimization
  - . The key difference between OSG and OAdagrad is that OAdagrad inherits ideas from Adagrad to construct variable metric based on history gradients information, while OSG only utilizes a fixed metric.

Assumption 2. *(i). There exists* $G > 0$ *and* $\delta > 0$ *such that* $\|T(\mathbf{z};\xi)\|_2 \leq G$, $\|T(\mathbf{z};\xi)\|_\infty \leq \delta$ *for all* $\mathbf{z}$ *almost surely.*

(ii). There exists a universal constant $D > 0$ *such that* $\|\mathbf{x}_k\|_2 \leq D/2$ *for* $k = 1, \ldots, N$, *and* $\|\mathbf{x}_*\|_2 \leq D/2$.

Optimistic Adagrad

Algorithm 2 Optimistic AdaGrad (OAdagrad)

Input: $\mathbf{z}_0 = \mathbf{x}_0 = 0$, $H_0 = \delta I$
for $k = 1, \ldots, N$ **do**
  $\mathbf{z}_k = \mathbf{x}_{k-1} - \eta H_{k-1}^{-1} \widehat{\mathbf{g}}_{k-1}$
  $\mathbf{x}_k = \mathbf{x}_{k-1} - \eta H_{k-1}^{-1} \widehat{\mathbf{g}}_k$
  Update $\widehat{\mathbf{g}}_{0:k} = [\widehat{\mathbf{g}}_{0:k-1}\ \widehat{\mathbf{g}}_k]$, $s_{k,i} = \|\widehat{\mathbf{g}}_{0:k,i}\|$, $i = 1, \ldots, d$ and set $H_k = \delta I + \text{diag}(s_{k-1})$
end for

- Similar to OSG where min variable and max variable are updated simultaneously.
- $\hat{g}_k$ represents the gradient at iteration k.
- $\hat{g}_{0:k}$ is the i-th row of the matrix obtained by concatenating the sub gradients from iteration 0 through k in the algorithm.

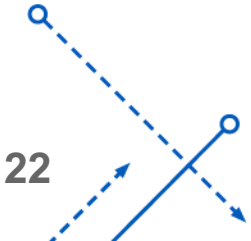
Theorem 2. *Suppose Assumption 1 and 2 hold. Suppose* $\|\widehat{\mathbf{g}}_{1:k,i}\|_2 \leq \delta k^{\alpha}$ *with* $0 \leq \alpha \leq 1/2$ *for every* $i = 1, \ldots, d$ *and every* $k = 1, \ldots, N$. *When* $\eta \leq \frac{\delta}{9L}$, *after running Algorithm 2 for* N *iterations, we have*

$$\frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\|T(\mathbf{z}_k)\|^2_{H^{-1}_{k-1}} \leq \frac{8D^2\delta^2(1+d(N-1)^{\alpha})}{\eta^2 N} + \frac{100\left(\sigma^2/m + d\left(2\delta^2 N^{\alpha} + G^2\right)\right)}{N}. \tag{6}$$

To make sure $\frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\|T(\mathbf{z}_k)\|^2_{H^{-1}_{k-1}} \leq \epsilon^2$, *the number of iterations is* $N = O\left(\epsilon^{-\frac{2}{1-\alpha}}\right)$.

Optimistic Adagrad

- Comparison with Alternating Adam and Optimistic Adam
  - OAdagrad updates the discriminator and generator simultaneously.
  - There is no convergence proof for Alternating Adam for non-convex non-concave problem
  - OAdagrad provides a theoretical justification of a special case of Optimistic Adam as OAdagrad naturally fits into the framework of Optimistic Adam



Contents

Experiments

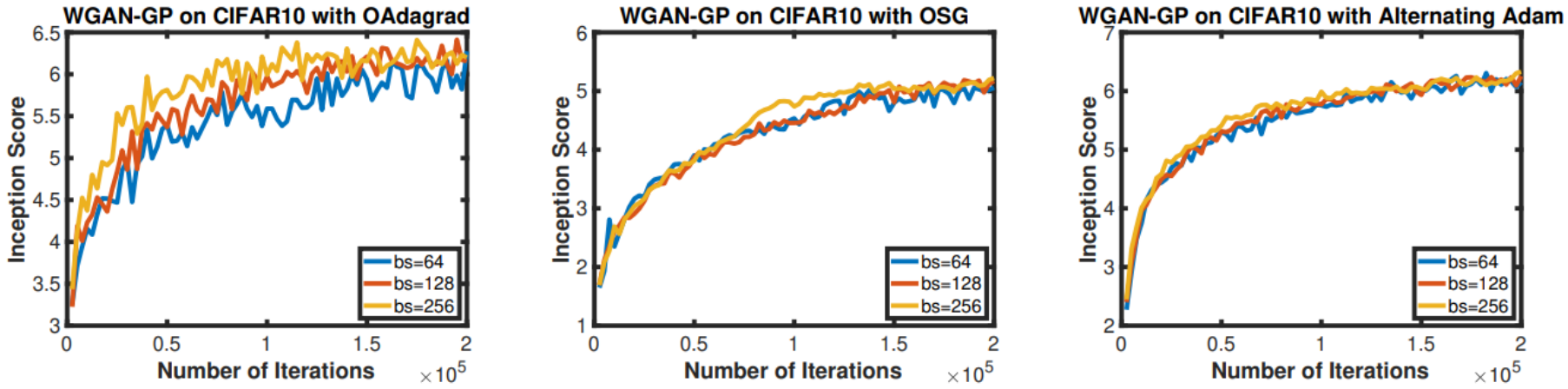


Figure 1: OAdagrad, OSG and Alternating Adam for WGAN-GP on CIFAR10 data

- OAdagrad performs better than OSG and Alternating Adam, and OAdagrad results in higher Inception Score.

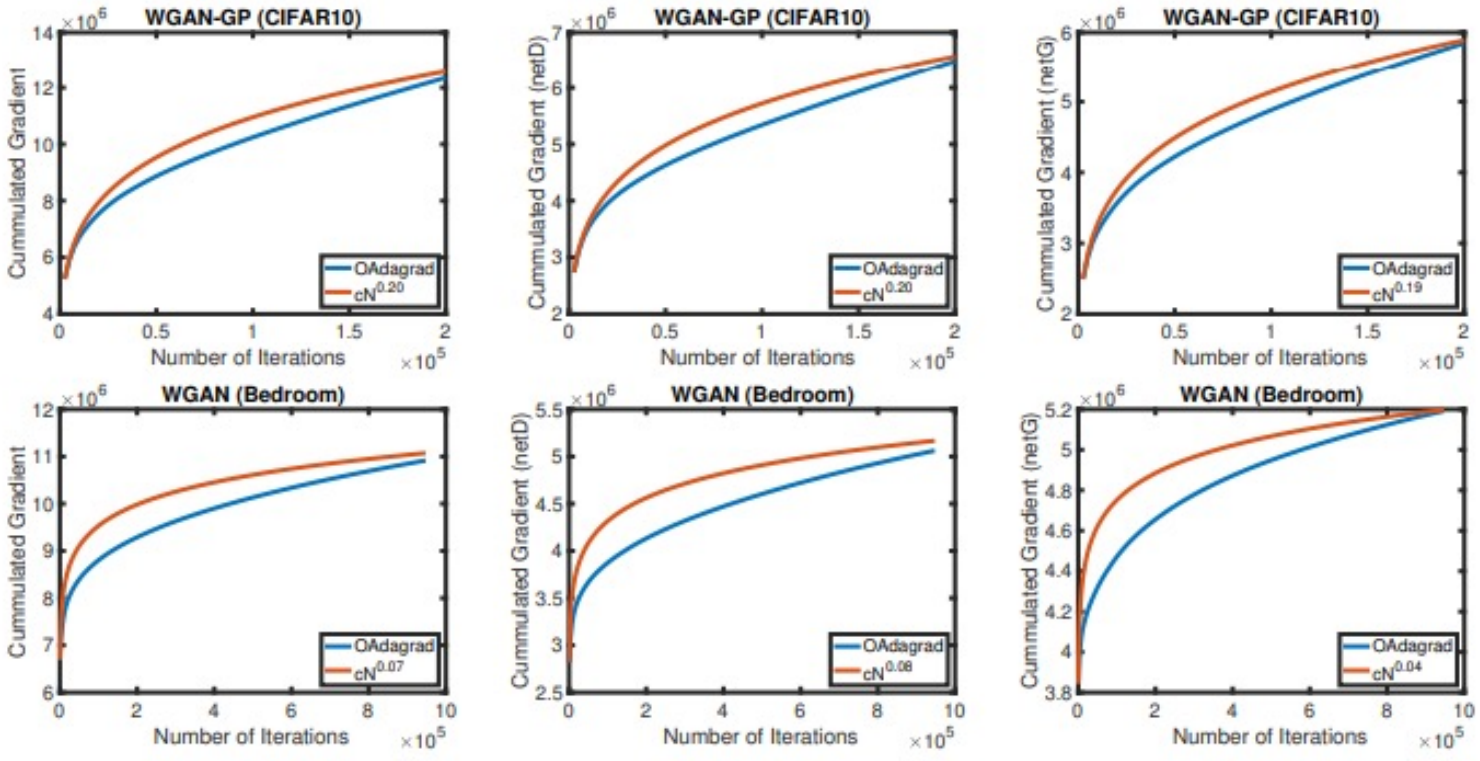
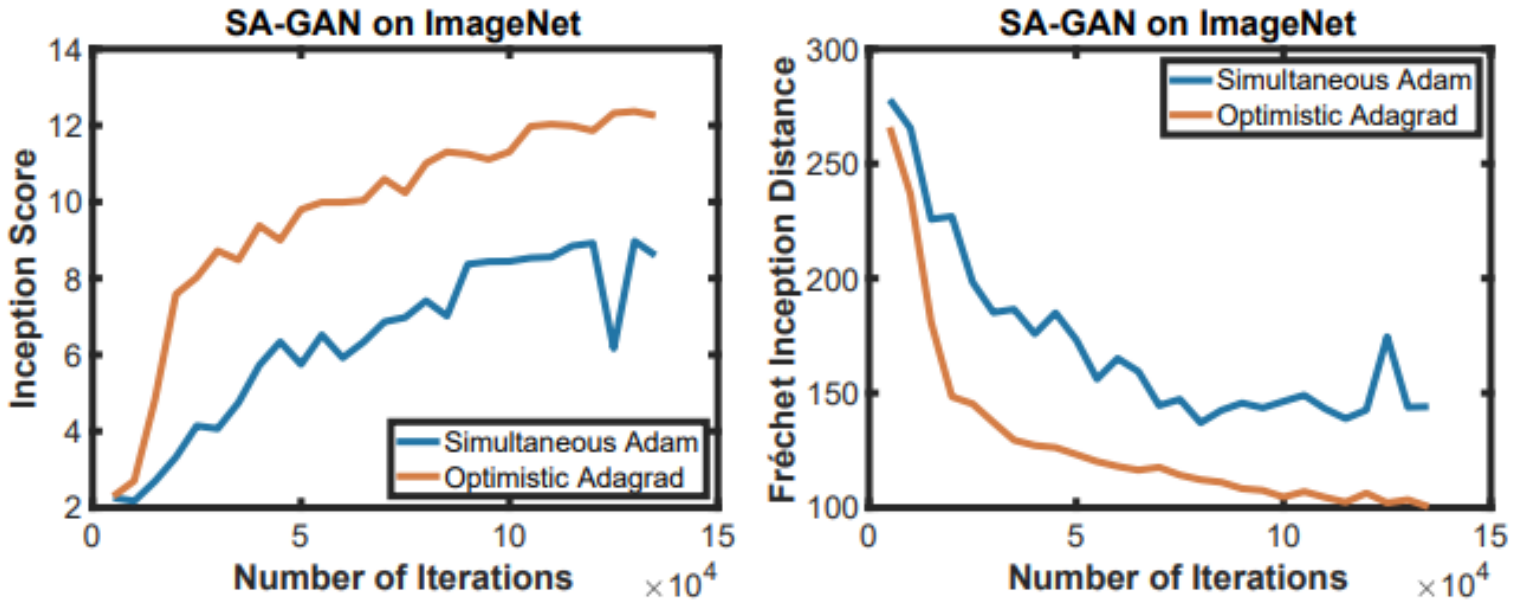


Figure 2: Cumulative Stochastic Gradient as a function of number of iterations, where netD and netG stand for the discriminator and generator respectively. The blue curve and red curve stand for the growth rate of the cummulative stochastic gradient for OAdagrad and its corresponding tightest polynomial growth upper bound, respectively.



(a) Inception Score (b) FID

Figure 3: Self-Attention GAN on ImageNet, with evaluation using Official TensorFlow Inception Score and Official TensorFlow FID. We see that OAdagard indeed outperforms Simultaneous Adam in terms of the (TensorFlow) Inception score (higher is better), and in terms of (TensorFlow) Fréchet Inception Distance (lower is better). We don't report here Alternating Adam since in our run it has collapsed.

Contents

Conclusion

- OAdagrad outperforms simultaneous Adam in quantitative metrics (IS and FID) and in sample quality generation.
- OAdagrad performs better than OSG and Alternating Adam, and OAdagrad results in higher IS on the CIFAR10 benchmark test.
- The algorithm is proven to enjoy faster adaptive convergence than its non-adaptive counterpart when the gradient is sparse.

References

- Liu, Mingrui, et al. Towards better understanding of adaptive gradient algorithms in generative adversarial nets. ICLR 2020.
- Komlósi, Sándor. (1999). On the Stampacchia and Minty variational inequalities. Generalized Convexity and Optimization for Economic and Financial Decisions.
- Tengfei000. (n.d.). *TENGFEI000/paper-reading*. GitHub.
- 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.