Optimization Methods and Software

ISSN: 1055-6788 (Print) 1029-4937 (Online) Journal homepage: https://www.tandfonline.com/loi/goms20

Training GANs with centripetal acceleration

Wei Peng, Yu-Hong Dai, Hui Zhang & Lizhi Cheng

To cite this article: Wei Peng, Yu-Hong Dai, Hui Zhang & Lizhi Cheng (2020): Training GANs with centripetal acceleration, Optimization Methods and Software, DOI: 10.1080/10556788.2020.1754414

To link to this article: https://doi.org/10.1080/10556788.2020.1754414

Published online: 16 Apr 2020.

OPTIMIZATION METHODS & SOFTWARE
https://doi.org/10.1080/10556788.2020.1754414

Taylor & Francis
Taylor & Francis Group

Training GANs with centripetal acceleration

Wei Peng[a], Yu-Hong Dai[b,c], Hui Zhang[d] and Lizhi Cheng[d]

[a]National Innovation Institute of Defense Technology, Chinese Academy of Military Science, Beijing, People's Republic of China; [b]Academy of Mathematics and Systems Science, Chinese Academy of Sciences, LSEC, ICMSEC, Beijing, People's Republic of China; [c]School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, People's Republic of China; [d]Department of Mathematics, National University of Defense Technology, Changsha, Hunan, People's Republic of China

ABSTRACT

Training generative adversarial networks (GANs) often suffers from cyclic behaviours of iterates. Based on a simple intuition that the direction of centripetal acceleration of an object moving in uniform circular motion is toward the centre of the circle, we present the *Simultaneous Centripetal Acceleration* (SCA) method and the *Alternating Centripetal Acceleration* (ACA) method to alleviate the cyclic behaviours. Under suitable conditions, gradient descent methods with either SCA or ACA are shown to be linearly convergent for bilinear games. Numerical experiments are conducted by applying ACA to existing gradient-based algorithms in a GAN setup scenario, which demonstrate the superiority of ACA.

ARTICLE HISTORY
Received 7 May 2019
Accepted 7 April 2020

AMS SUBJECT CLASSIFICATIONS
Primary: 97N60; 90C25;
Secondary: 90C06; 90C30

1. Introduction

The gradient method has attracted a lot of attention in machine learning in recent years. The earliest gradient method is called steepest descent method and can be dated back to Cauchy (1847). The properties of the steepest descent method have been well studied for strictly convex quadratic functions and it is known that it is very slowly in practice and produces zigzags. In 1988, Barzilai and Borwein [3] proposed the two-point stepsize gradient method, called Barzilai-Borwein gradient method, which is non-monotone but performs significantly better than the steepest descent method. To look for a new gradient method which is monotone and also numerically efficient, Yuan [23] provided a new stepsize for the gradient method based on the finite termination property for the two-dimensional strictly convex functions. This directly leads to the proposition of the Dai-Yuan gradient method (see the formula (5.3) in [5]), which is monotone and even performs slightly better than the Barzilai-Borwein gradient method for strictly convex quadratic functions. Nowadays, there have been a lot of studies following the above works. In this paper, we shall look for some new gradient methods for training Generative Adversarial Nets.

CONTACT Yu-Hong Dai dyh@lsec.cc.ac.cn Academy of Mathematics and Systems Science, Chinese Academy of Sciences, LSEC, ICMSEC, Beijing 100190, People's Republic of China; School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, People's Republic of China

Generative Adversarial Nets (GANs) [10] are recognized as powerful generative models, which have successfully been applied to various fields such as image generation [12], representation learning [19] and super resolution [21]. The idea behind GANs is an adversarial game between a generator network (G-net) and a discriminator network (D-net). The G-net attempts to generate synthetic data from some noise to deceive the D-net while the D-net tries to discern between the synthetic data and the real data. The original GANs can be formulated as the min-max problem:

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim p_{data}}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]. \tag{1}$$

Though GANs are appealing, they are often hard to train. The main difficulty might be the associated gradient vector field rotating around a Nash equilibrium due to the existence of imaginary components in the Jacobian eigenvalues [15], which results in the limit oscillatory behaviours. There are a series of studies focussing on developing fast and stable methods of training GANs. Using the Jacobian, consensus optimization [15] diverts gradient updates to the descent direction of the field magnitudes. More essentially, a differential game can always be decomposed into a *potential game* and a *Hamiltonian game* [2]. Potential games have been intensively studied [17] because gradient decent methods converge in these games. Hamiltonian games obey a conservation law such that iterates generated by gradient descent are likely to cycle or even diverge in these games. Therefore, Hamiltonian components might be the cause of cycling when gradient descent methods are applied. Based on the observations, the *Symplectic Gradient Adjustment* (SGA) method [2] modifies the associated vector field to guide the iterates to *cross the curl* of the Hamiltonian component of a differential game. Gemp and Mahadevan [7] also uses the similar technique to cross the curl such that rotations are alleviated. By augmenting the Follow-the-Regularized-Leader algorithm with an l_2 regularizer [20] by adding an *optimistic* predictor of the next iteration gradient, *Optimistic Mirror Descent* (OMD) methods are presented in [6] and analysed in [8,13,14,16]. Mertikopoulos *et al.* [14] extends the OMD from the perspective of Bregman proximal gradient methods. The Hamiltonian dynamic system has also been studied in [2,7,14], which we will follow in our analysis. The negative momentum is employed in [9] to deplete the kinetic energy of the cyclic motion such that iterates would fall towards the centre. It is also observed in [9] that the alternating version of the negative momentum method is more stable. Another important issue is to figure out the relationship between the optimal points of GANs and stable stationary points of an algorithm under the nonconvex-noncave settings. Jin *et al.* [11] proposed a modified gradient method called CESP, where its stable points are equivalent to locally optimal saddles. [1] points out the global minimax point may be intractable. Therefore, the definition of a proper locally optimal point is fundamental. Relations among local Nash equilibrium, local minimax points and stable points are also revealed by [1].

Our idea is motivated by two aspects. Firstly and intuitively, we use the fact that the direction of centripetal acceleration of an object moving in uniform circular motion points to the centre of the circle, which might guide iterates to cross the curl and escape from cycling traps. To avoid confusion, the word of acceleration does not mean the speeding up of convergence, but comes from the physics concept of centripetal acceleration. Secondly, we try to find a method to approximate the dynamics of consensus optimization or SGA to cross the curl without computing the Jacobian, which can reduce computational costs.

Then we were inspired to present the centripetal acceleration methods, which can be used to adjust gradients in various methods such as SGD, RMSProp [22] and Adam [4]. For stability and effectiveness, we are also motivated by [9] to study the alternating scheme, which could even work in a notorious GAN setup scenario.

The main contributions are as follows:

(1) From two different perspectives, we present centripetal acceleration methods to alleviate the cyclic behaviours in training GANs. Specifically, we propose the *Simultaneous Centripetal Acceleration* (SCA) method and the *Alternating Centripetal Acceleration* (ACA) method.
(2) For bilinear games, which are purely adversarial, we prove that gradient descent with either SCA or ACA is linearly convergent under suitable conditions.
(3) Primary numerical simulations are conducted in a GAN setup scenario, which show that the centripetal acceleration is useful while combining several gradient-based algorithms.

Outline. The rest of the paper is organized as follows. In Section 2, we present simultaneous and alternating centripetal acceleration methods and discuss them with closely related works. In Section 3, focussing on bilinear games, we prove the linear convergence of gradient descent combined with the two centripetal acceleration methods. In Section 4, we conduct numerical experiments to test the effectiveness of centripetal acceleration methods. Section 5 concludes the paper.

2. Centripetal acceleration methods

A differentiable two-player game involves two loss functions $l_1(\theta,\phi)$ and $l_2(\theta,\phi)$ defined over a parameter space $\Omega_\theta \times \Omega_\phi$. Player 1 tries to minimize the loss l_1 while player 2 attempts to minimize the loss l_2. The goal is to find a local Nash equilibrium of the game, i.e. a pair $(\bar\theta,\bar\phi)$ with the following two conditions holding in a neighbourhood of $(\bar\theta,\bar\phi)$:

$$\bar\theta \in \arg\min_{\theta} l_1(\theta,\bar\phi), \quad \bar\phi \in \arg\min_{\phi} l_2(\bar\theta,\phi).$$

The derivation of problem (1) leads to a two-player game. The G-net is parameterized as $G(\cdot\,;\theta)$ while the D-net is parameterized as $D(\cdot\,;\phi)$. Then the problem becomes to find a local Nash equilibrium:

$$\bar\theta \in \arg\min_{\theta} \left\{V(\theta,\bar\phi)\right\}, \quad \bar\phi \in \arg\min_{\phi} \left\{-V(\bar\theta,\phi)\right\}, \tag{2}$$

where

$$V(\theta,\phi) = \mathbb{E}_{x\sim p_{data}}[\log D(x;\phi)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z;\theta);\phi))]. \tag{3}$$

The simultaneous gradient descent method in training GANs [18] is

$$\theta_{t+1} = \theta_t - \alpha\nabla_\theta V(\theta_t,\phi_t), \quad \phi_{t+1} = \phi_t + \alpha\nabla_\phi V(\theta_t,\phi_t).$$

The alternating version is

$$\theta_{t+1} = \theta_t - \alpha\nabla_\theta V(\theta_t,\phi_t), \quad \phi_{t+1} = \phi_t + \alpha\nabla_\phi V(\theta_{t+1},\phi_t).$$

However, directly applying gradient descent even fails to approach the saddle point in a toy model (See Figure 2 in Section 4). By applying the *Simultaneous Centripetal Acceleration* (SCA) method, which will be explained later, to adjust gradients, we obtain the method of *Gradient descent with SCA* (Grad-SCA):

$$G_\theta = \nabla_\theta V(\theta_t, \phi_t) + \frac{\beta_1}{\alpha_1}(\nabla_\theta V(\theta_t, \phi_t) - \nabla_\theta V(\theta_{t-1}, \phi_{t-1})), \tag{4}$$

$$\theta_{t+1} = \theta_t - \alpha_1 G_\theta, \tag{5}$$

$$G_\phi = \nabla_\phi V(\theta_t, \phi_t) + \frac{\beta_2}{\alpha_2}(\nabla_\phi V(\theta_t, \phi_t) - \nabla_\phi V(\theta_{t-1}, \phi_{t-1})), \tag{6}$$

$$\phi_{t+1} = \phi_t + \alpha_2 G_\phi. \tag{7}$$

It can be seen that the gradient decent scheme is still employed in (5) and (7), while the gradients in (4) and (6) are adjusted by adding the directions of centripetal acceleration simultaneously. If adjusting the gradients by the *Alternating Centripetal Acceleration* (ACA) method, we obtain the following method of *Gradient descent with ACA* (Grad-ACA):

$$G_\theta = \nabla_\theta V(\theta_t, \phi_t) + \frac{\beta_1}{\alpha_1}(\nabla_\theta V(\theta_t, \phi_t) - \nabla_\theta V(\theta_{t-1}, \phi_{t-1})), \tag{8}$$

$$\theta_{t+1} = \theta_t - \alpha_1 G_\theta, \tag{9}$$

$$G_\phi = \nabla_\phi V(\theta_{t+1}, \phi_t) + \frac{\beta_2}{\alpha_2}(\nabla_\phi V(\theta_{t+1}, \phi_t) - \nabla_\phi V(\theta_t, \phi_{t-1})), \tag{10}$$

$$\phi_{t+1} = \phi_t + \alpha_2 G_\phi. \tag{11}$$

Grad-ACA also employs simple gradient descent steps but adjusts the gradients by adding the directions of centripetal acceleration alternatively. Nevertheless, the idea of centripetal acceleration can also be applied to other gradient-based methods, resulting in more efficient algorithms. For example, the RMSProp algorithm [22] with ACA, abbreviated by RMSProp-ACA, performs well in our numerical experiments (see Section 4.2).

The basic intuition behind employing centripetal acceleration is shown in Figure 1. Consider the uniform circular motion. Let ∇V_t denote the instantaneous velocity at time t. Then the centripetal acceleration $\lim_{\delta t \to 0}(\nabla V_{t+\delta t} - \nabla V_t)/\delta t$ points to the origin. The cyclic behaviour around a Nash equilibrium might be similar to the circular motion around the origin. Therefore, the centripetal acceleration provides a direction, along which the iterates can approach the target more quickly. Then the approximated centripetal acceleration term $(\nabla V(\theta_t, \phi_t) - \nabla V(\theta_{t-1}, \phi_{t-1}))$ is applied to gradient descent as illustrated in Grad-SCA.

The proposed centripetal acceleration methods are also inspired by the dynamics of consensus optimization. In a Hamiltonian game, the associated vector field ∇V conserves the Hamiltonian's level sets because $\langle \nabla V, \nabla \|\nabla V\|^2 \rangle = 0$, which prevents iterates from approaching the equilibrium where $\|\nabla V\| = 0$. To illustrate the similarity between centripetal acceleration methods and consensus optimization in Hamiltonian games, we consider the n-player differential game where each player has a loss function $l_i(w_1, w_2, \ldots, w_n)$ for $i = 1, 2, \ldots, n$. Then the simultaneous gradient is

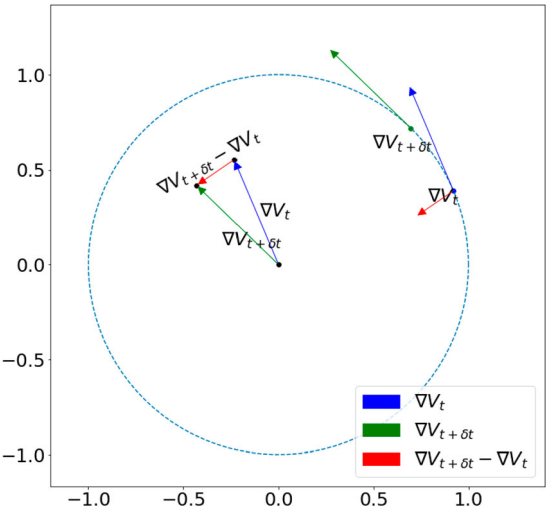

Figure 1. The basic intuition of centripetal acceleration methods.

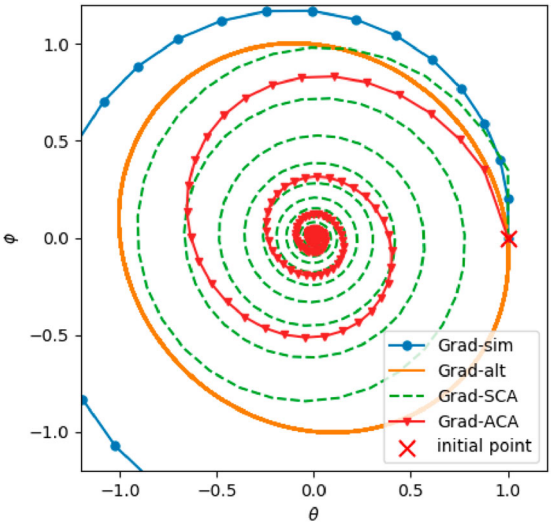

Figure 2. The effects of Grad-SCA and Grad-ACA in the simple bilinear game. Simultaneous gradient descent ($\alpha = 0.1, \beta = 0$) diverges while the alternating gradient descent ($\alpha = 0.1, \beta = 0$) keeps the iterates running on a closed trajectory. Instead, both Grad-SCA and Grad-ACA ($\alpha = 0.1, \beta = 0.3$) converge to the origin linearly and the alternating version seems faster.

$\xi(w_1, w_2, \ldots, w_n) := (\nabla_{w_1} l_1, \nabla_{w_2} l_2, \ldots, \nabla_{w_n} l_n)$. The Jacobian of ξ is

$$J := \begin{bmatrix} \nabla_{w_1 w_1} l_1 & \nabla_{w_1 w_2} l_1 & \cdots & \nabla_{w_1 w_n} l_1 \\ \nabla_{w_2 w_1} l_2 & \nabla_{w_2 w_2} l_2 & \cdots & \nabla_{w_1 w_2} l_2 \\ \cdots & \cdots & \cdots & \cdots \\ \nabla_{w_n w_1} l_n & \nabla_{w_n w_2} l_n & \cdots & \nabla_{w_n w_n} l_n \end{bmatrix}. \tag{12}$$

Let $w := (w_1, w_2, \ldots, w_n)$. Then the iteration scheme of consensus optimization is

$$w_{k+1} = w_k - \alpha(\xi_k + \beta J_k^{\mathrm{T}} \xi_k) \tag{13}$$

and the corresponding continuous dynamics has the form:

$$\frac{\mathrm{d}w}{\mathrm{d}t} = -(I + \beta J^{\mathrm{T}})\xi. \tag{14}$$

When β is small, the dynamics approximates

$$\frac{\mathrm{d}w}{\mathrm{d}t} = -(I - \beta J^{\mathrm{T}})^{-1}\xi. \tag{15}$$

By rearranging the order, we obtain

$$\frac{\mathrm{d}w}{\mathrm{d}t} = -\xi + \beta J^{\mathrm{T}} \frac{\mathrm{d}w}{\mathrm{d}t}. \tag{16}$$

Since the game is assumed to be Hamiltonian, i.e. $J = -J^{\mathrm{T}}$, the dynamic equation (16) becomes

$$\frac{\mathrm{d}w}{\mathrm{d}t} = -\xi - \beta J \frac{\mathrm{d}w}{\mathrm{d}t}. \tag{17}$$

Note that $J(\mathrm{d}w/\mathrm{d}t) = \mathrm{d}\xi/\mathrm{d}t$. Then (17) is equivalent to

$$\frac{\mathrm{d}w}{\mathrm{d}t} = -\xi - \beta \frac{\mathrm{d}\xi}{\mathrm{d}t}. \tag{18}$$

Discretizing the equation with stepsize α, we obtain

$$w_{t+1} = w_t - \alpha\xi_t - \beta(\xi_t - \xi_{t-1}), \tag{19}$$

which is exactly Grad-SCA. Furthermore, in Hamiltonian games, the dynamics of consensus optimization and SGA that plugs into gradient descent algorithms (Grad-SGA) are essentially the same. Therefore, the presented Grad-SCA could be regarded as a Jacobian-free approximation of consensus optimization or Grad-SGA.

Related works. Taking $\alpha_1 = \alpha_2 = \beta_1 = \beta_2 = \alpha$ in Grad-SCA (4)–(7), the centripetal acceleration scheme reduces to OMD [6], which has the following form:

$$\begin{aligned}
\theta_{t+1} &= \theta_t - 2\alpha\nabla_\theta V(\theta_t, \phi_t) + \alpha\nabla_\theta V(\theta_{t-1}, \phi_{t-1}),\\
\phi_{t+1} &= \phi_t + 2\alpha\nabla_\phi V(\theta_t, \phi_t) - \alpha\nabla_\phi V(\theta_{t-1}, \phi_{t-1}).
\end{aligned}$$

Very recently, from the perspective of generalizing OMD, [16] presented schemes similar to Grad-SCA and they studied its convergence under a unified proximal method framework. However, OMD is motivated by predicting the next iteration gradient to be the current gradient optimistically. Although the scheme of OMD coincides with Grad-SCA, we must stress that the motivations are essentially different and result in totally distinct parameter selection strategies. Due to the similar dynamics, the presented methods inherit parameter selection strategies of consensus optimization and SGA. For example, in the second experiment in Section 4, we take $\alpha_1 = \alpha_2 = 5 \times 10^{-4}$ and $\beta_1 = \beta_2 = 0.5$. The magnitude

of β is quite larger than α instead of an equality. Moreover, we analyze the alternating form (Grad-ACA) (8)–(11) and employed RMSProp-ACA in the numerical experiments. Therefore, the presented methods are not trivial generalizations of OMD and the idea of centripetal acceleration is quite useful.

Another similar scheme [8] is to extrapolate the gradient from the past:

$$\theta_{t+1/2} = \theta_t - \alpha\nabla_\theta V(\theta_{t-1/2}, \phi_{t-1/2}), \quad \phi_{t+1/2} = \phi_t + \alpha\nabla_\phi V(\theta_{t-1/2}, \phi_{t-1/2}),$$
$$\theta_{t+1} = \theta_t - \alpha\nabla_\theta V(\theta_{t+1/2}, \phi_{t+1/2}), \quad \phi_{t+1} = \phi_t + \alpha\nabla_\phi V(\theta_{t+1/2}, \phi_{t+1/2}).$$

It can be rewritten as

$$\theta_{t+1/2} = \theta_{t-1/2} - 2\alpha\nabla_\theta V(\theta_{t-1/2}, \phi_{t-1/2}) + \alpha\nabla_\theta V(\theta_{t-3/2}, \phi_{t-3/2}),$$
$$\phi_{t+1/2} = \phi_{t-1/2} + 2\alpha\nabla_\phi V(\theta_{t-1/2}, \phi_{t-1/2}) - \alpha\nabla_\phi V(\theta_{t-3/2}, \phi_{t-3/2})$$

which is equivalent to OMD. The algorithm may also be closely related to the predictive methods with the following form:

$$\theta_{t+1/2} = \theta_t - \alpha\nabla V(\theta_t, \phi_t), \quad \phi_{t+1/2} = \phi_t + \alpha\nabla V(\theta_t, \phi_t),$$
$$\theta_{t+1} = \theta_t - \beta\nabla V(\theta_{t+1/2}, \phi_{t+1/2}), \quad \phi_{t+1} = \phi_t + \beta\nabla V(\theta_{t+1/2}, \phi_{t+1/2}).$$

A unified framework to analyze OMD and predictive methods is presented in [13].

Last but not least, our idea of using alternating scheme comes from negative momentum methods [9], which suggests alternating forms might be more stable and effective in practice.

3. Linear convergence for bilinear games

In this section, we focus on the convergence of Grad-SCA and Grad-ACA in the bilinear game:

$$\min_{\theta\in\mathbb{R}^d}\max_{\phi\in\mathbb{R}^p} \theta^{\mathrm{T}}A\phi + \theta^{\mathrm{T}}b + c^{\mathrm{T}}\phi, \quad A \in \mathbb{R}^{d\times p}, \ \ b \in \mathbb{R}^d, \ \ c \in \mathbb{R}^p. \tag{20}$$

Any stationary point (θ^*, ϕ^*) of the game satisfies the first order conditions:

$$A\phi^* + b = 0 \tag{21}$$

$$A^{\mathrm{T}}\theta^* + c = 0. \tag{22}$$

It is obvious that a stationary point exists if and only if b is in the range of A and c is in the range of A^{T}. We suppose that such a pair (θ^*, ϕ^*) exists. Without loss of generality, we shift (θ, ϕ) to $(\theta - \theta^*, \phi - \phi^*)$. Then the problem is reformulated as:

$$\min_{\theta\in\mathbb{R}^d}\max_{\phi\in\mathbb{R}^p} \theta^{\mathrm{T}}A\phi, \quad A \in \mathbb{R}^{d\times p}. \tag{23}$$

In the following two subsections, we analyze convergence properties of Grad-SCA and Grad-ACA, respectively. Technique details are postponed to appendices. In the following subsections and the corresponding appendices, for a matrix $A \in \mathbb{R}^{d\times d}$, let $\mathrm{Sp}(A)$ denote the collection of its eigenvalues. Given vectors $v_1, v_2, \ldots, v_k \in \mathbb{R}^d$, let $\mathrm{Span}(v_1, v_2, \ldots, v_k)$ denote the subspace spanned by these vectors. For a given subspace $S \subset \mathbb{R}^d$, let $P_S(\cdot)$ denote the projection onto S.

3.1. Linear convergence of Grad-SCA

For the bilinear game, Grad-SCA is specified as

$$\theta_{t+1} = \theta_t - \alpha_1 A\phi_t - \beta_1(A\phi_t - A\phi_{t-1}), \tag{24}$$

$$\phi_{t+1} = \phi_t + \alpha_2 A^{\mathrm{T}}\theta_t + \beta_2(A^{\mathrm{T}}\theta_t - A^{\mathrm{T}}\theta_{t-1}). \tag{25}$$

Define the matrix $F_1 \in \mathbb{R}^{2p+2d}$ as

$$F_1 := \begin{bmatrix} I_d & -(\alpha_1+\beta_1)A & 0 & \beta_1 A \\ (\alpha_2+\beta_2)A^{\mathrm{T}} & I_p & -\beta_2 A^{\mathrm{T}} & 0 \\ I_d & 0 & 0 & 0 \\ 0 & I_p & 0 & 0 \end{bmatrix}. \tag{26}$$

It is obvious that $[\theta_{t+1}, \phi_{t+1}, \theta_t, \phi_t]^{\mathrm{T}} = F_1[\theta_t, \phi_t, \theta_{t-1}, \phi_{t-1}]^{\mathrm{T}}$, where (θ_t, ϕ_t) are generated by (24) and (25). For simplicity, we suppose that A is square and nonsingular in Propositions 3.2 and 3.3 and Corollary 3.4. Then we prove the linear convergence for a general matrix A in Proposition 3.5 and Corollary 3.6. We will employ the following well-known lemma to illustrate the linear convergence.

Lemma 3.1: *Suppose that $F \in \mathbb{R}^{p\times p}$ has the spectral radius $\rho(F) < 1$. Then the iterative system $x_{k+1} = Fx_k$ converges to 0 linearly. Explicitly, $\forall\ \varepsilon > 0$, there exists a constant $C > 0$ such that*

$$\|x_t\| \le C\left(\rho(F) + \varepsilon\right)^t. \tag{27}$$

Proposition 3.2: *Suppose that A is square and nonsingular. The eigenvalues of F_1 are the roots of the fourth order polynomials:*

$$\lambda^2(1-\lambda)^2 + (\lambda(\alpha_2+\beta_2) - \beta_2)(\lambda(\alpha_1+\beta_1) - \beta_1)\zeta, \quad \zeta \in \mathrm{Sp}(A^{\mathrm{T}}A), \tag{28}$$

where $\mathrm{Sp}(\cdot)$ *denotes the collection of all eigenvalues.*

Next, we consider cases when $\alpha_1 = \alpha_2 = \alpha$ and $\beta_1 = \beta_2 = \beta$.

Proposition 3.3: *Suppose that A is square and nonsingular. Then $\Delta_t := \|\theta_t\|^2 + \|\phi_t\|^2 + \|\theta_{t+1}\|^2 + \|\phi_{t+1}\|^2$ is linearly convergent to 0 if α and β satisfy*

$$0 < \alpha + \beta \le \frac{1}{\sqrt{\lambda_{\max}(A^{\mathrm{T}}A)}}, \quad |\alpha - \beta| \le \frac{\sqrt{\lambda_{\min}(A^{\mathrm{T}}A)}|\alpha+\beta|^2}{10}, \tag{29}$$

where $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ denote the largest and the smallest eigenvalues, respectively.

Consider the special case when Grad-SCA reduces to OMD. Then we have the following corollary. The corollary is slightly weaker than the existing result [13, Lemma 3.1].

Corollary 3.4: *Suppose that A is square and nonsingular. If $\alpha_1 = \alpha_2 = \beta_1 = \beta_2 = \alpha$ and $0 < \alpha \leq \frac{1}{\sqrt{\lambda_{\max}(A^{\mathrm{T}}A)}}$, then Δ_t is linearly convergent, i.e. $\forall \varepsilon > 0$, there exists $C > 0$ such that*

$$\Delta_t \leq C\left(\varepsilon + \sqrt{\tfrac{1}{2} + \tfrac{1}{2}\sqrt{1 - \alpha^2\lambda_{\min}(A^{\mathrm{T}}A)}}\right)^{2t}.$$

Now we do not assume A to be square and nonsingular ($d \geq p$). Instead, suppose A has rank r and the SVD decomposition is $A = UDV^{\mathrm{T}}$, where $D = \mathrm{diag}\{\sigma_1, \sigma_2, \ldots, \sigma_r, 0, \ldots, 0\} \in \mathbb{R}^{p\times p}$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$, $U \in \mathbb{R}^{d\times p}$ and $V \in \mathbb{R}^{p\times p}$. Denote by M the null space of A, which means $M = \{x \in \mathbb{R}^p \mid Ax = 0\}$, and by N the null space of A^{T}. Note that any $(\tilde{\theta}, \tilde{\phi}) \in N \times M$ is a stationary point and we define

$$\Delta_t^P := \|\theta_{t+1} - P_N(\theta_0)\|^2 + \|\theta_t - P_N(\theta_0)\|^2 + \|\phi_{t+1} - P_M(\phi_0)\|^2 + \|\phi_t - P_M(\phi_0)\|^2,$$

where $P_N(\cdot)$ denotes the orthogonal projection onto N while $P_M(\cdot)$ denotes the orthogonal projection onto M.

Proposition 3.5: *Suppose that $0 < \alpha + \beta \leq 1/\sigma_1$ and $|\alpha - \beta|/|\alpha + \beta|^2 \leq 0.1\sigma_r$. Then Δ_t^P is linearly convergent.*

With the analogous analysis, we have the following result for OMD.

Corollary 3.6: *If $\alpha_1 = \alpha_2 = \beta_1 = \beta_2 = \alpha$ and $0 < \alpha \leq 1/\sigma_1$, then Δ_t^P is linearly convergent, i.e. $\forall\, \varepsilon > 0$, there exists a constant $C > 0$ such that*

$$\Delta_{t+1}^P \leq C\left(\varepsilon + \sqrt{\tfrac{1}{2} + \tfrac{1}{2}\sqrt{1 - \alpha^2\sigma_r^2}}\right)^{2t}.$$

3.2. Linear convergence of Grad-ACA

In this subsection, we consider Grad-ACA for the bilinear game,

$$\theta_{t+1} = \theta_t - \alpha_1 A\phi_t - \beta_1(A\phi_t - A\phi_{t-1}), \tag{30}$$

$$\phi_{t+1} = \phi_t + \alpha_2 A^{\mathrm{T}}\theta_{t+1} + \beta_2(A^{\mathrm{T}}\theta_{t+1} - A^{\mathrm{T}}\theta_t). \tag{31}$$

The update of ϕ_{t+1} can be rewritten as:

$$\phi_{t+1} = \phi_t + (\alpha_2 + \beta_2)A^{\mathrm{T}}(\theta_t - \alpha_1 A\phi_t - \beta_1(A\phi_t - A\phi_{t-1})) - \beta_2 A^{\mathrm{T}}\theta_t.$$

Thus we define the matrix

$$F_2 := \begin{bmatrix} I & -(\alpha_1+\beta_1)A & 0 & \beta_1 A \\ \alpha_2 A^{\mathrm{T}} & I - (\alpha_1+\beta_1)(\alpha_2+\beta_2)A^{\mathrm{T}}A & 0 & (\alpha_2+\beta_2)\beta_1 A^{\mathrm{T}}A \\ I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \end{bmatrix}, \tag{32}$$

which immediately follows that $[\theta_{t+1}, \phi_{t+1}, \theta_t, \phi_t]^{\mathrm{T}} = F_2[\theta_t, \phi_t, \theta_{t-1}, \phi_{t-1}]^{\mathrm{T}}$.

Proposition 3.7: *Suppose that A is square and nonsingular. Consider the special case where $\beta_1 = 0, \alpha_1 = \alpha_2 = \beta_2 = \alpha$. If $0 < \alpha \leq 1/\sqrt{2\lambda_{\max}(A^{\mathrm{T}}A)}$, then $\Delta_t := \|\theta_t\|^2 + \|\phi_t\|^2$ is linearly convergent to 0, i.e. there exists a constant $C > 0$ such that*

$$\Delta_t \leq C\left(1 - \alpha^2\lambda_{\min}(A^{\mathrm{T}}A) + \alpha^4\lambda_{\min}(A^{\mathrm{T}}A)^2\right)^{2t}.$$

Next, we do not assume A to be square and nonsingular. Employing the SVD decomposition $A = UDV^{\mathrm{T}}$ and with the same techniques employed in Proposition 3.5, we have

Corollary 3.8: *Consider the special case where $\beta_1 = 0, \alpha_1 = \alpha_2 = \beta_2 = \alpha$. If $0 < \alpha \leq \sqrt{2}/2\sigma_1$, Then $\Delta_t^P := \|\theta_t - P_N(\theta_0)\|^2 + \|\phi_t - P_M(\phi_0)\|^2$ is linearly convergent, i.e. there exists a constant $C > 0$ such that*

$$\Delta_t^P \leq C(1 - \alpha^2\sigma_r^2 + \alpha^4\sigma_r^4)^{2t},$$

which implies that (θ_t, ϕ_t) linearly converges to the stationary point $(P_N(\theta_0), P_M(\phi_0))$.

4. Numerical simulation

4.1. A simple bilinear game

In the first experiment, we tested Grad-SCA and Grad-ACA on the following bilinear game

$$\min_{\theta\in\mathbb{R}} \max_{\phi\in\mathbb{R}} \theta \cdot \phi. \tag{33}$$

The unique stationary point is $(\theta^*, \phi^*) = (0, 0)$. The behaviours of the methods are presented in Figure 2. Pure gradient descent steps do not converge to the origin in this simple

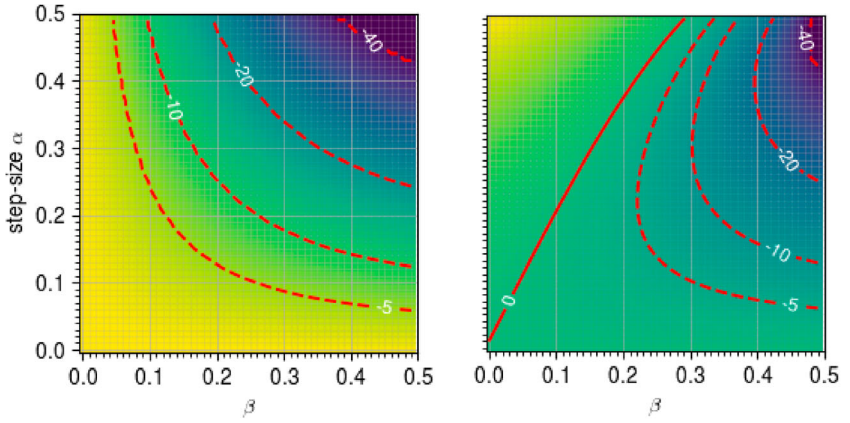

Figure 3. Parameter selection in the simple bilinear game. We test Grad-SCA and Grad-ACA with varying parameters $(\alpha, \beta) \in (0, 0.5] \times (0, 0.5]$. Each grid point represents the logarithm of the squared distance to the origin after 500 iterations. Note that the colormaps are different between the two images. Grad-ACA (left) converges in the entire parameter box while Grad-SCA (right) might diverge if the step-size α is much larger than β. In this simple experiment, a larger β seems more preferred. Particularly, when $\alpha = \beta$, the Grad-SCA reduces to OMD.

game. However, with centripetal acceleration methods, both Grad-SCA and Grad-ACA converge to the origin.

We compared the effects of various step-sizes and acceleration coefficients in both simultaneous and alternating cases. Figure 3 suggests that the alternating methods are preferable.

4.2. Mixture of Gaussians

The theoretical convergence of Grad-SCA and Grad-ACA on the simple bilinear model might imply their potential applications in training GANs. Motivated by the viewpoint, we provide the following example. In the second simulation[1], we established a toy GAN model to compare several methods on learning eight Gaussians with standard deviation 0.04. The ground truth is shown in Figure 4.

Both the generator and the discriminator networks have four fully connected layers of 256 neurones. The sigmoid function layer is appended to the discriminator to normalize the output. Each of the four layers is activated by a ReLU layer. The generator has two

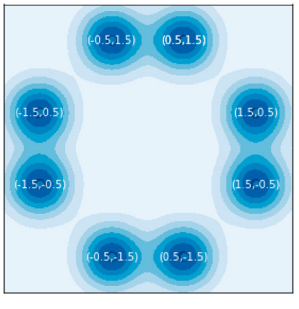

Figure 4. Kernel density estimation on 2560 samples of the ground truth.

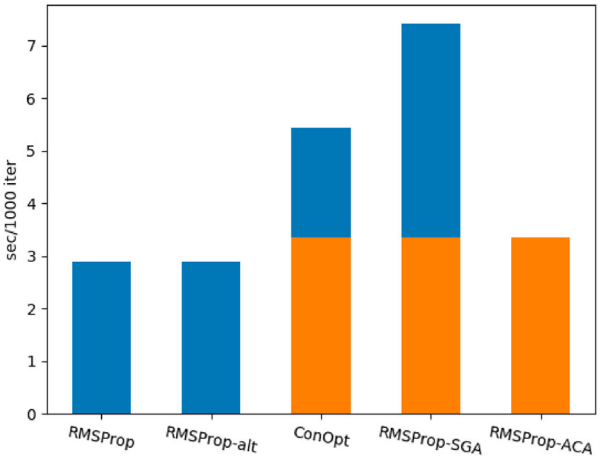

Figure 5. Comparison among several algorithms on the mixture of Gaussians. Five methods are included in the comparison. Each row displays one method and each column shows samples generated by the G-net at 1000, 2000, 4000, 8000 iterations respectively.

output neurones to represent a generated point while the discriminator has one output which judges a sample. The random noise input for the generator is a 16-D Gaussian. To be explicit, the generator network $G(\cdot\,;\theta) : \mathbb{R}^{16} \to \mathbb{R}^2$ is parameterized by the network weight collection θ and the discriminator network $D(\cdot;\phi) : \mathbb{R}^2 \to \mathbb{R}$ is parameterized by ϕ. In the kth iteration, we sample $\{x_i^{(k)}\}_{i=1,2,\ldots,256}$ from the Gaussian mixture distribution p_{data} and sample $\{z_i^{(k)}\}_{i=1,2,\ldots,256}$ from the Gaussian noise distribution p_z. Then we perform

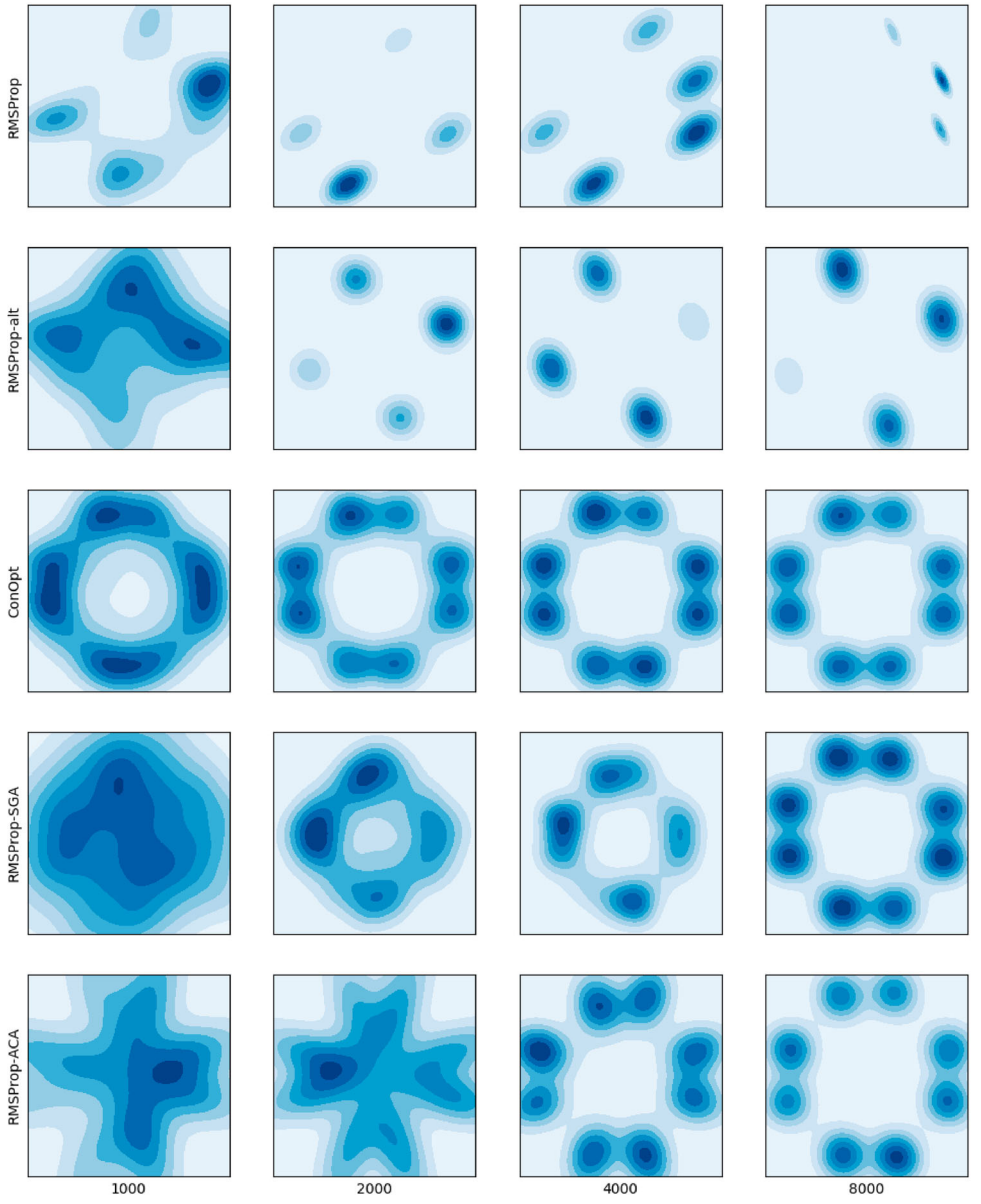

Figure 6. Time consuming comparison. RMSProp-ACA consumes slightly more time than RMSProp. However, it takes far less time than ConOpt and RMSProp-SGA.

one step of the gradient descent operation on the weights (θ, ϕ) according to the first-order information provided by

$$V^{(k)}(\theta,\phi)|_{(\theta_k,\phi_k)} = \frac{1}{256}\left(\sum_{i=1}^{256}\log D(x_i^{(k)};\phi) + \sum_{i=1}^{256}\log(1 - D(G(z_i^{(k)};\theta);\phi))\right). \quad (34)$$

We conducted the experiment on a server equipped with CPU i7 4790, GPU Titan Xp, 16GB RAM as well as TensorFlow (version 1.12) and Python (version 3.6.7).

We compared the results of several algorithms as shown in Figure 5 . Five methods are included in the comparison:

(1) *RMSProp*: Simultaneous RMSPropOptimizer (learning rate: $\alpha = 5 \times 10^{-4}$) provided by TensorFlow.
(2) *RMSProp-alt*: Alternating RMSPropOptimizer (learning rate: $\alpha = 5 \times 10^{-4}$).
(3) *ConOpt*: Consensus optimizer ($h = 10^{-4}, \gamma = 1$) [15].
(4) *RMSProp-SGA*: Symplectic gradient adjusted RMSPropOptimizer with sign alignment (learningrate $= 10^{-4}, \xi = 1$) [2].
(5) *RMSProp-ACA*: RMSPropOptimizer with alternating centripetal acceleration method ($\alpha = 5 \times 10^{-4}, \beta = 0.5$). The parameter is selected via a coarse grid-search.

To stress the effectiveness brought by parameter selection and alternating strategy regardless of the similar form with OMD, we also tested OMD on this simulation with searching a range of parameters (See Appendix 2).

The centripetal acceleration methods have extra computation costs on computing the difference between successive gradients as well as storage costs to maintain previous gradients. The consensus optimization and SGA require extra computations on the Jacobian related steps. Figure 6 shows a time consuming comparison. From these comparisons, RMSProp-ACA seems competitive to other methods.

5. Conclusion

In this paper, to alleviate the difficulty in finding a local Nash equilibrium in a smooth two-player game, we were inspired to present several gradient-based methods, including Grad-SCA and Grad-ACA, which employ centripetal acceleration. The proposed methods can easily be plugged into other gradient-based algorithms like SGD, Adam or RMSProp in both simultaneous or alternating ways. From the theoretical viewpoint, we proved that both Grad-SCA and Grad-ACA have linear convergence for bilinear games under suitable conditions. We found that in a simple bilinear game, centripetal acceleration makes iterates converge to the Nash equilibrium stably; these examples also suggest that alternating methods are more preferred than simultaneous ones. In the GAN setup numerical simulations, we showed that the RMSProp-ACA can be competitive to consensus optimization and symplectic gradient adjustment methods.

However, we only consider the deterministic bilinear games theoretically and limited numerical simulations. In practical training of GANs or its variants, the associated games are much more complicated due to the randomness of computation, the online procedure and non-convexity. These issues still need further detailed studies.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

Yu-Hong Dai research is supported by the Chinese NSF grand (Nos. 11631013, 11991021, 11971372 and 11991020) and Beijing Academy of Artificial Intelligence (BAAI). Hui Zhang research is supported by the Chinese NSF grants (Nos. 11501569 and 61571008).

Note

1. The code is available at https://github.com/dynames0098/GANsTrainingWithCenAcc

Notes on contributors

Wei Peng received the PhD degree from the National University of Defense Technology, Changsha, China, in 2019. After his graduation, he worked in the National Innovation Institute of Defense Technology, Chinese Academy of Military Science, Beijing, China, where he is currently an Assistant Research Scientist. His interests include nonlinear optimization and mathematical foundation of deep learning.

Yu-Hong Dai received his bachelor's degree in applied mathematics from the Beijing Institute of Technology, Beijing, China, in 1992. He then studied nonlinear optimization in the Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences (CAS), Beijing, China, and received his Ph. in 1997. After his graduation, he worked in the Academy of Mathematics and Systems Science (AMSS) of CAS and became a Full Professor in 2006. Currently, he is the assistant president of AMSS of CAS and vice president of the Operations Research Society of China. His research interests include continuous optimization, integer programming, and applied optimization.

Hui Zhang received the PhD degree from the National University of Defense Technology, Changsha, China, in 2014, where he is currently an Associate Professor with the Department of Mathematics. His research interests include Variational Analysis and Sparse Optimization.

Lizhi Cheng received the PhD degree from the National University of Defense Technology, Changsha, China, in 2002, where he is currently a Professor with the Department of Mathematics. He has authored or co-authored over 100 technical articles in different fields, including optimization and control, image processing, and wavelet analysis. His research interests include the mathematical foundation of signal analysis and wavelet analysis with applications to image compression.

References

[1] L. Adolphs, H. Daneshmand, A. Lucchi, and T. Hofmann, Local saddle point optimization: A curvature exploitation approach. *arXiv preprint arXiv:1805.05751*, 2018.

[2] D. Balduzzi, S. Racaniere, J. Martens, J. Foerster, K. Tuyls, and T. Graepel, *The mechanics of n-player differentiable games*, International Conference on Machine Learning, Stockholmsmässan, Stockholm Swede, 2018, pp. 363–372.

[3] J. Barzilai and J.M. Borwein, *Two-point step size gradient methods*, IMA J. Numer. Anal. 8(1–2) (1988), pp. 141–148.

[4] T. Chilimbi, Y. Suzue, J. Apacible, and K. Kalyanaraman, *Project adam: Building an efficient and scalable deep learning training system*, Proceedings of the 11th USENIX conference on Operating Systems Design and Implementation, USENIX Association, Broomfield, CO, 2014, pp. 571–582.

[5] Y.-H. Dai and Y. Yuan, *Analysis of monotone gradient methods*, J. Ind. Manage. Optim. 1(2) (2005), pp. 181–192.
[6] C. Daskalakis, A. Ilyas, V. Syrgkanis, and H. Zeng, *Training gans with optimism*, Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 2018.
[7] I. Gemp and S. Mahadevan, *Global convergence to the equilibrium of gans using variational inequalities*, preprint (2018). Available at arXiv:1808.01531.
[8] G. Gidel, H. Berard, P. Vincent, and S. Lacoste-Julien, *A variational inequality perspective on generative adversarial nets*, preprint (2018). Available at arXiv:1802.10551.
[9] G. Gidel, R.A. Hemmat, M. Pezeshki, G. Huang, R. Lepriol, S. Lacoste-Julien, and I. Mitliagkas, *Negative momentum for improved game dynamics*, preprint (2018). Available at arXiv:1807.04740.
[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets*, Advances in Neural Information Processing Systems 27, Palais des Congrès de Montréal, Montréal, Canada, 2014, pp. 2672–2680.
[11] C. Jin, P. Netrapalli, and M.I. Jordan, *Minmax optimization: Stable limit points of gradient descent ascent are locally optimal*, preprint (2019). Available at arXiv:1902.00618.
[12] T. Karras, T. Aila, S. Laine, and J. Lehtinen, *Progressive growing of gans for improved quality, stability, and variation*, preprint (2017). Available at arXiv:1710.10196.
[13] T. Liang and J. Stokes, *Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks*, preprint (2018). Available at arXiv:1802.06132.
[14] P. Mertikopoulos, H. Zenati, B. Lecouat, C.-S. Foo, V. Chandrasekhar, and G. Piliouras, *Mirror descent in saddle-point problems: Going the extra (gradient) mile*, preprint (2018). Available at arXiv:1807.02629.
[15] L. Mescheder, S. Nowozin, and A. Geiger, *The numerics of gans*, Advances in Neural Information Processing Systems 30, Long Beach Convention Center, Long Beach, 2017, pp. 1825–1835.
[16] A. Mokhtari, A. Ozdaglar, and S. Pattathil, *A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach*, preprint (2019). Available at arXiv:1901.08511.
[17] D. Monderer and L.S. Shapley, *Potential games*, Games Econ. Behav. 14(1) (1996), pp. 124–143.
[18] S. Nowozin, B. Cseke, and R. Tomioka, *f-gan: Training generative neural samplers using variational divergence minimization*, Advances in Neural Information Processing Systems 29, Centre Convencions Internacional Barcelona, Barcelona, Spain, 2016, pp. 271–279.
[19] A. Radford, L. Metz, and S. Chintala, *Unsupervised representation learning with deep convolutional generative adversarial networks*, preprint (2015). Available at arXiv:1511.06434.
[20] S. Shalev-Shwartz, *Online learning and online convex optimization*, Foundations and Trendső in Machine Learning 4(2) (2012), pp. 107–194.
[21] W. Shi, C. Ledig, Z. Wang, L. Theis, and F. Huszar, *Super resolution using a generative adversarial network, March 15 2018*, US Patent App. 15/706,428.
[22] T. Tieleman and G. Hinton, *Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude*, COURSERA 4(2) (2012), pp. 26–31.
[23] Y. Yuan, *A new stepsize for the steepest descent method*, J. Comput. Math. 24(2) (2006), pp. 149–156.

Appendix 1. Proofs in Section 3

A.1 Proof of Proposition 3.2

Proof: The characteristic polynomial of the matrix (26) is

$$\det\begin{pmatrix} I_d-\lambda I_d & -(\alpha_1+\beta_1)A & 0 & \beta_1 A \\ (\alpha_2+\beta_2)A^{\mathrm{T}} & I_p-\lambda I_p & -\beta_2 A^{\mathrm{T}} & 0 \\ I_d & 0 & -\lambda I_d & 0 \\ 0 & I_p & 0 & -\lambda I_p \end{pmatrix}, \tag{A1}$$

which is equivalent to

$$\det\begin{pmatrix}\lambda(1-\lambda)I_d & \lambda(\alpha_1+\beta_1)A-\beta_1 A\\ -\lambda(\alpha_2+\beta_2)A^{\mathrm{T}}+\beta_2 A^{\mathrm{T}} & \lambda(1-\lambda)I_p\end{pmatrix}. \tag{A2}$$

Since A is nonsingular and square, then 0 or 1 can not be the roots of A2. Then the roots of (A2) must be the roots of

$$\det\left(\lambda(1-\lambda)I_p+\frac{1}{\lambda(1-\lambda)}\left(\lambda(\alpha_2+\beta_2)-\beta_2\right)\left(\lambda(\alpha_1+\beta_1)-\beta_1\right)A^{\mathrm{T}}A\right). \tag{A3}$$

It follows that the eigenvalues of F_1 must be the roots of the fourth order polynomials:

$$\lambda^2(1-\lambda)^2+(\lambda(\alpha_2+\beta_2)-\beta_2)(\lambda(\alpha_1+\beta_1)-\beta_1)\zeta,\quad \zeta\in \mathrm{Sp}(A^{\mathrm{T}}A).$$

■

A.2 Proof of Proposition 3.3

Proof: Given an eigenvalue λ of F_1, using Proposition 3.2, we have

$$\left(\lambda^2-\lambda-\mathrm{i}(\lambda(\alpha+\beta)-\beta)\sqrt{\zeta}\right)\left(\lambda^2-\lambda+\mathrm{i}(\lambda(\alpha+\beta)-\beta)\sqrt{\zeta}\right)=0,\quad \zeta\in\mathrm{Sp}(A^{\mathrm{T}}A). \tag{A4}$$

Denote $s:=\alpha\sqrt{\zeta}+\beta\sqrt{\zeta}$ and $t:=\alpha\sqrt{\zeta}-\beta\sqrt{\zeta}$. Then the four roots of (A4) are

$$\lambda_1^{\pm}=\frac{1+\mathrm{i}s\pm\sqrt{1+2\mathrm{i}t-s^2}}{2},$$
$$\lambda_2^{\pm}=\frac{1-\mathrm{i}s\pm\sqrt{1-2\mathrm{i}t-s^2}}{2}.$$

Note that for a given complex number z, the absolute value of the real part of $z^{1/2}$ is $\sqrt{(|z|+R(z))/2}$ and the absolute value of the imaginary part of $z^{1/2}$ is $\sqrt{(|z|-R(z))/2}$. Therefore, since $s\le 1$, all real parts of $\lambda_i^{\pm}(i=1,2)$ lie in the interval $[-\mathcal{R},\mathcal{R}]$, where

$$\mathcal{R}=\frac{1}{2}\sqrt{\frac{\sqrt{(1-s^2)^2+4t^2}+1-s^2}{2}}+\frac{1}{2} \tag{A5}$$

and all imaginary parts of $\lambda_i^{\pm}(i=1,2)$ lie in the interval $[-\mathcal{I},\mathcal{I}]$, where

$$\mathcal{I}=\frac{1}{2}\sqrt{\frac{\sqrt{(1-s^2)^2+4t^2}-1+s^2}{2}}+\frac{s}{2}. \tag{A6}$$

Using the inequality

$$\sqrt{x+y}\le\sqrt{x}+\frac{y}{2\sqrt{x}},\quad (x>0,y\ge 0), \tag{A7}$$

we have

$$\mathcal{R}\le\frac{1}{2}\sqrt{1-s^2+\frac{t^2}{1-s^2}}+\frac{1}{2}, \tag{A8}$$

$$\mathcal{I}\le\frac{s}{2}+\frac{|t|}{2\sqrt{1-s^2}}. \tag{A9}$$

Next, we discuss s in $(0,1/\sqrt{2}]$ and $(1/\sqrt{2},1]$ separately.

(1). In the first case, we suppose $0 < s \le 1/\sqrt{2}$. Since $|\alpha - \beta|/(\alpha+\beta)^2 \le 0.1\sqrt{\zeta}$ for all $\zeta \in \mathrm{Sp}(A^{\mathrm{T}}A)$, we have

$$|t| \le \frac{s^2}{10}.$$

Noting that $s^2/2 \le 1 - \sqrt{1-s^2}$, we obtain

$$|t| \le \frac{1-\sqrt{1-s^2}}{5}. \tag{A10}$$

Combining $s \le 1/\sqrt{2}$ and (A10) yields

$$\begin{aligned} |t| &\le \frac{2(1-\sqrt{1-s^2})(1-s^2)}{5} \\ &\le \frac{(1-\sqrt{1-s^2})(1-s^2)}{2\sqrt{1-s^2}+\frac{1}{2}}, \end{aligned}$$

which follows that

$$\begin{aligned} 1 &\ge \frac{|t|}{\sqrt{1-s^2}} + \sqrt{1-s^2} + \frac{|t|}{2(1-s^2)} + \frac{|t|}{\sqrt{1-s^2}} \\ &\ge \frac{t^2}{1-s^2} + \sqrt{1-s^2} + \frac{t^2}{2(1-s^2)^{3/2}} + \frac{s|t|}{\sqrt{1-s^2}} \qquad \text{(A11)} \\ &\ge \frac{t^2}{1-s^2} + \sqrt{1-s^2+\frac{t^2}{1-s^2}} + \frac{s|t|}{\sqrt{1-s^2}}. \qquad \text{(A12)} \end{aligned}$$

The inequality (A11) follows by the fact that $|t|/\sqrt{1-s^2} \le s/\sqrt{1-s^2} \le 1$ and the inequality (A12) uses (A7). The inequality above is equivalent to

$$\left(\frac{1}{2}\sqrt{1-s^2+\frac{t^2}{1-s^2}} + \frac{1}{2}\right)^2 + \left(\frac{s}{2} + \frac{|t|}{2\sqrt{1-s^2}}\right)^2 \le 1.$$

Using (A8) and (A9), we obtain

$$\rho(F_1) \le \sqrt{\mathcal{R}^2+\mathcal{I}^2} \le 1. \tag{A13}$$

Note that the equality of (A7) holds if and only if $y = 0$. Thus the equality of (A13) implies $t = 0$ and $s = 0$. Since $s > 0$, we have the strict inequality $\rho(F_1) < 1$, which leads to the linear convergence of Δ_t.

(2). In the second case, assume $1/\sqrt{2} < s \le 1$. Since $t \le s^2/10 \le 0.1$, using (A5) and (A6) directly, we have

$$\rho(F_1) \le \sqrt{\mathcal{R}^2+\mathcal{I}^2} < 1, \tag{A14}$$

which yields the linear convergence. ■

A.3 Proof of Corollary 3.4

Proof: For the special cases, we have $t = 0$ and $0 < s \le 1$. From (A8) and (A9), we obtain

$$\begin{aligned} \rho(F_1) \le \sqrt{\mathcal{R}^2+\mathcal{I}^2} &= \tfrac{1}{2}\sqrt{\left((1-s^2)+1+2\sqrt{1-s^2}\right)+s^2} \\ &= \sqrt{\tfrac{1}{2}+\tfrac{1}{2}\sqrt{1-s^2}} \\ &\le \sqrt{\tfrac{1}{2}+\tfrac{1}{2}\sqrt{1-\alpha^2\lambda_{\min}(A^{\mathrm{T}}A)}} < 1. \end{aligned}$$

From Lemma 3.1 it follows that Δ_t is linearly convergent. ■

A.4 Proof of Proposition 3.5

Proof: Using the SVD decomposition $A = UDV^{\mathrm{T}}$, we have

$$U^{\mathrm{T}}\theta_{t+1} = U^{\mathrm{T}}\theta_t - \alpha DV^{\mathrm{T}}\phi_t - \beta(DV^{\mathrm{T}}\phi_t - DV^{\mathrm{T}}\phi_{t-1}),$$
$$V^{\mathrm{T}}\phi_{t+1} = V^{\mathrm{T}}\phi_t + \alpha DU^{\mathrm{T}}\theta_t + \beta(DU^{\mathrm{T}}\theta_t - DU^{\mathrm{T}}\theta_{t-1}).$$

According to the definition of the diagonal matrix D, the $(r+1)$th component to pth components of $DU^{\mathrm{T}}\theta_t$ and $DV^{\mathrm{T}}\phi_t$ are zeros. Therefore, we focus on the leading r components of $U^{\mathrm{T}}\theta_t$ and $V^{\mathrm{T}}\phi_t$, denoted by $[U^{\mathrm{T}}\theta_t]_{1:r}$ and $[V^{\mathrm{T}}\phi_t]_{1:r}$ respectively. Let D_r be the matrix composed of the leading r rows and columns of D. Then we have

$$[U^{\mathrm{T}}\theta_{t+1}]_{1:r} = [U^{\mathrm{T}}\theta_t]_{1:r} - \alpha D_r[V^{\mathrm{T}}\phi_t]_{1:r} - \beta(D_r[V^{\mathrm{T}}\phi_t]_{1:r} - D_r[V^{\mathrm{T}}\phi_{t-1}]_{1:r}), \tag{A15}$$
$$[V^{\mathrm{T}}\phi_{t+1}]_{1:r} = [V^{\mathrm{T}}\phi_t]_{1:r} + \alpha D_r[U^{\mathrm{T}}\theta_t]_{1:r} + \beta(D_r[U^{\mathrm{T}}\theta_t]_{1:r} - D_r[U^{\mathrm{T}}\theta_{t-1}]_{1:r}). \tag{A16}$$

Define

$$\begin{aligned}\Delta_t^r :&= \|[U^{\mathrm{T}}\theta_t]_{1:r}\|^2 + \|[U^{\mathrm{T}}\theta_{t+1}]_{1:r}\|^2 + \|[V^{\mathrm{T}}\phi_t]_{1:r}\|^2 + \|[V^{\mathrm{T}}\phi_{t+1}]_{1:r}\|^2 \\ &= \|\theta_t - P_N(\theta_t)\|^2 + \|\theta_{t+1} - P_N(\theta_{t+1})\|^2 + \|\phi_t - P_M(\phi_t)\|^2 + \|\phi_{t+1} - P_M(\phi_{t+1})\|^2. \end{aligned} \tag{A17}$$

The equality (A17) holds due to $N^{\perp} = \mathrm{Span}\{u_1, u_2, \ldots, u_r\}$ and $M^{\perp} = \mathrm{Span}\{v_1, v_2, \ldots, v_r\}$. Since D_r is square and nonsingular, applying Proposition 3.3 to (A15) and (A16), we have that Δ_t^r is linearly convergent. Recalling (24), (25), for all $u \in N, v \in M$, we have

$$u^{\mathrm{T}}\theta_{t+1} = u^{\mathrm{T}}\theta_t,$$
$$v^{\mathrm{T}}\phi_{t+1} = v^{\mathrm{T}}\phi_t,$$

which implies $P_N(\theta_t) = P_N(\theta_0)$ and $P_M(\phi_t) = P_M(\phi_0)$ for all $t \geq 0$. Then we have $\Delta_t^r = \Delta_t^P$ for all $t \geq 0$. Thus Δ_t^P is also linearly convergent. ■

A.5 Proof of Proposition 3.7

Proof: The iterations (30) and (31) are simplified to

$$\theta_{t+1} = \theta_t - \alpha A\phi_t,$$
$$\phi_{t+1} = (I - 2\alpha^2 A^{\mathrm{T}}A)\phi_t + \alpha A^{\mathrm{T}}\theta_t.$$

Then the matrix F_2 reduces to

$$\tilde{F}_2 = \begin{bmatrix} I & -\alpha A \\ I - 2\alpha^2 A^{\mathrm{T}}A & \alpha A^{\mathrm{T}} \end{bmatrix}$$

with $[\theta_{t+1}, \phi_{t+1}]^{\mathrm{T}} = \tilde{F}_2[\theta_t, \phi_t]^{\mathrm{T}}$ holding. Eigenvalues of $\tilde{F}_2$ satisfy

$$\lambda^2 - (1 - 2\alpha^2\zeta)\lambda - \alpha^2\zeta = 0, \quad \zeta \in \mathrm{Sp}(A^{\mathrm{T}}A).$$

Let $a := \alpha\sqrt{\zeta}$. Then the two roots are

$$\lambda = \frac{1 - 2a^2 \pm \sqrt{1 + 4a^4}}{2}. \tag{A18}$$

For all $a \in (0, 1/\sqrt{2})$, applying (A7) to (A18) we have

$$|\lambda| < 1 - a^2 + a^4.$$

Note that $f(x) = 1 - x^2 + x^4$ is monotone decreasing on $(0, 1/\sqrt{2})$. Then

$$\rho(\tilde{F}_2) < 1 - \alpha^2\lambda_{\min}(A^{\mathrm{T}}A) + \alpha^4\lambda_{\min}(A^{\mathrm{T}}A)^2 < 1.$$

Therefore, Δ_t is linearly convergent. ■

Appendix 2. Performance of OMD on mixture of Gaussians

For the performance of OMD in the second experiment in Section 4, we search the learning rates on the grid from 0.00002 to 50. The result is as Figure A1 shows. With varying learning rates, OMD combined with RMSProp suffers from mode collapse and fails to recover the Gaussian mixture even after 20k iterations.

Figure A1. Performance of OMD. From left to right, top to bottom, the stepsize α successively takes values from [2E-5, 5E-5, 1E-4, 2E-4, 5E-4, 1E-3, 2E-3, 5E-3, 1E-2, 2E-2, 5E-2, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50].