Efficient Saddle Point Optimization for Modern Machine Learning

Prédoc III - Gauthier Gidel

Jury:
Président : Yoshua Bengio
Membre : Ioannis Mitliagkas
Directeur : Simon Lacoste-Julien

Outline

1. Introduction on Saddle point optimization,
   Games and Variational Inequalities.

2. Frank-Wolfe Algorithm for Saddle Point problems.

3. Negative Momentum for improved game dynamics.

4. A Variational inequality perspective on GANs.

5. Future Work.

NB: All the citations in this talk are at the end of the slides.

Slides available on my website: *http://gauthiergidel.github.io*

Saddle point optimization, Games and Variational Inequalities.

Based on [Gidel et al. 2017], [Gidel et al. 2018a] and [Gidel et al. 2018b]

Game dynamics are ~~weird~~
fascinating

Start with optimization dynamics

Optimization

$$\boldsymbol{\theta} \in \operatorname*{argmin}_{\boldsymbol{\theta} \in \Theta} \mathcal{L}(\boldsymbol{\theta})$$

Smooth, **differentiable** cost function, L

→ Looking for stationary (fixed) points (gradient is 0)

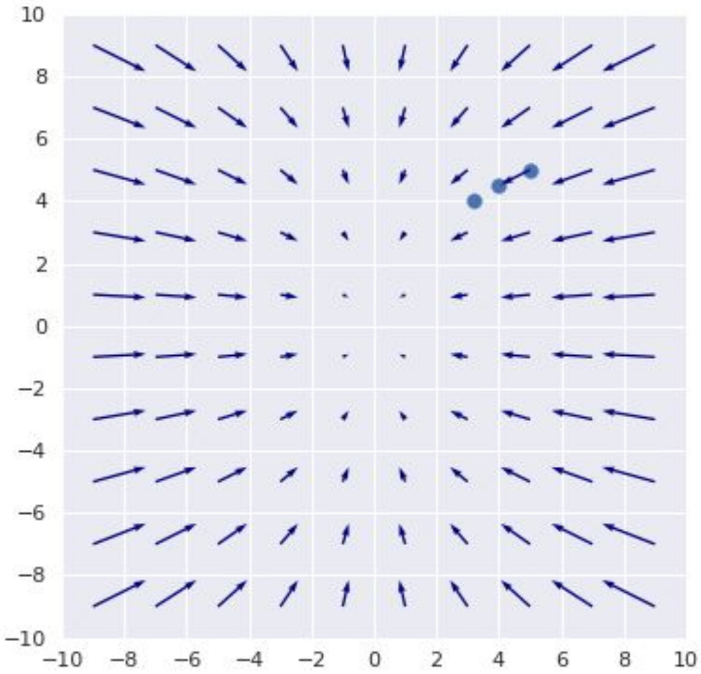
→ Gradient descent

Optimization

Conservative vector field →

Gradient based dynamics

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla \mathcal{L}(\boldsymbol{\theta}_t)$$



Mila Université de Montréal McGill

Gauthier Gidel,
Predoc III , November 28, 2018

Saddle point problems

$$\min_{\boldsymbol{\theta} \in \Theta} \max_{\boldsymbol{\phi} \in \Phi} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$$

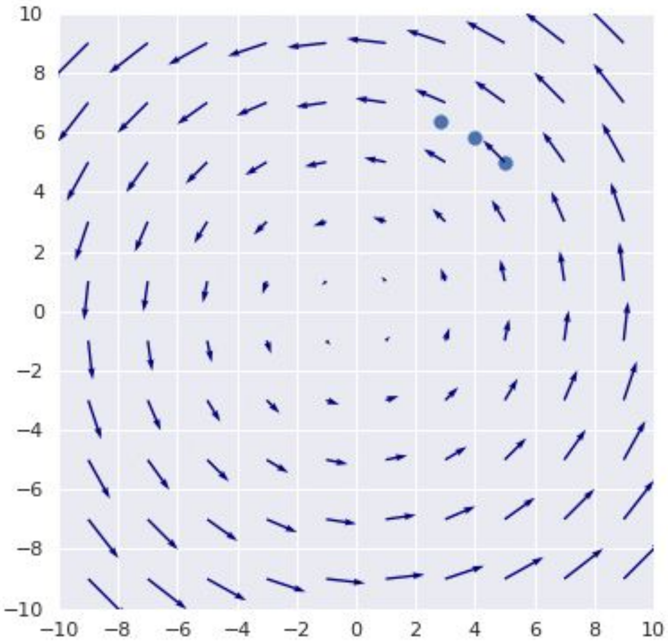
Smooth, **differentiable** cost function,

→ Looking for stationary (fixed) points
(gradients are 0)

→ Gradient ~~descent~~ **method.**

Saddle point problems

Non-Conservative vector field →



Gradient based dynamics:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_t, \boldsymbol{\phi}_t)$$

$$\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + \eta \nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\theta}_t, \boldsymbol{\phi}_t)$$

Mila Université de Montréal McGill

Minmax training is ~~hard~~ different !

Minmax training is ~~hard~~ different !

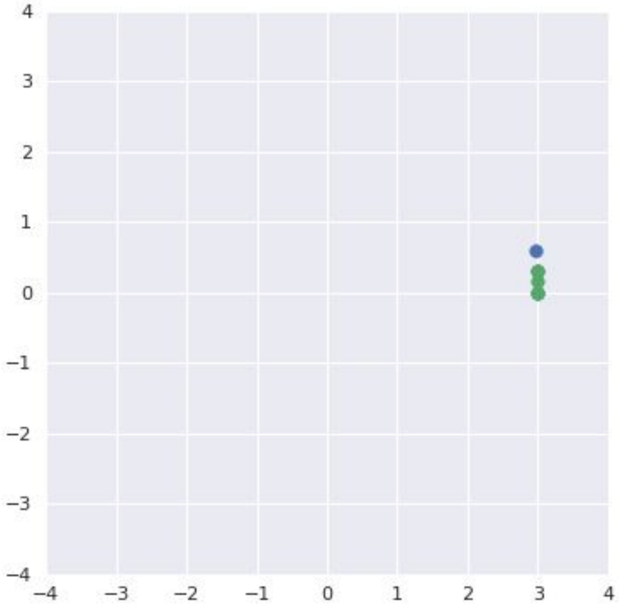
(You can replace “minmax” with two-player games)

"Minmax Training is Hard …"

Dynamics:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_t, \boldsymbol{\phi}_t)$$

$$\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + \eta \nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\theta}_t, \boldsymbol{\phi}_t)$$



Bilinear saddle point = Linear in θ and ϕ
⇒ "Cycling behavior" (see right).

Example: WGAN [Arjovky et al. 2017] with **linear** discriminator and generator

$$\min_{\theta} \max_{\phi, ||f_\phi||_L \leq 1} \phi^T \mathbb{E}_{x \sim p_{\mathcal{D}}}[x] - \phi^T \theta \mathbb{E}_{z \sim p_{\mathcal{Z}}}[z]$$

Gauthier Gidel,
Predoc III , November 28, 2018

Multi-player Games

Two-player Games

Player 1 Player 2

$$\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}\in\Theta} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}, \boldsymbol{\varphi}^*) \quad \text{and} \quad \boldsymbol{\varphi}^* \in \arg\min_{\boldsymbol{\varphi}\in\Phi} \mathcal{L}^{(\boldsymbol{\varphi})}(\boldsymbol{\theta}^*, \boldsymbol{\varphi})$$

Zero-sum game if: $\mathcal{L}^{(\boldsymbol{\theta})} = -\mathcal{L}^{(\boldsymbol{\varphi})}$ also called *Saddle Point* (SP).

Example: WGAN formulation [Arjovsky et al. 2017]

$$\min_{\theta} \max_{\phi, ||f_\phi||_L \leq 1} \underbrace{\mathbb{E}_{x\sim p_{\mathcal{D}}}[f_\phi(x)] - \mathbb{E}_{z\sim p_{\mathcal{Z}}}[f_\phi(g_\theta(z))]}_{\mathcal{L}^{(\boldsymbol{\theta})} = -\mathcal{L}^{(\boldsymbol{\varphi})}}$$

Two-player Games

Player 1 Player 2

$$\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}\in\Theta} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}, \boldsymbol{\varphi}^*) \quad \text{and} \quad \boldsymbol{\varphi}^* \in \arg\min_{\boldsymbol{\varphi}\in\Phi} \mathcal{L}^{(\boldsymbol{\varphi})}(\boldsymbol{\theta}^*, \boldsymbol{\varphi})$$

Non zero-sum game if we **do not** have: $\mathcal{L}^{(\boldsymbol{\theta})} = -\mathcal{L}^{(\boldsymbol{\varphi})}$

Example: Non-saturating GAN: [Goodfellow et al. 2014]

Loss of Generator Loss of Discriminator

$$\min_{\theta} -\mathbb{E}_{z\sim p_{\mathcal{Z}}}[\log(D_\phi(G_\theta(z)))] \qquad \max_{\phi} \mathbb{E}_{x\sim p_{\mathcal{D}}}[\log(D_\phi(x))] + \mathbb{E}_{z\sim p_{\mathcal{Z}}}[\log(1 - D_\phi(G_\theta(z)))]$$

Two-player Games

Player 1 | Player 2

$$\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}\in\Theta} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}, \boldsymbol{\varphi}^*) \quad \text{and} \quad \boldsymbol{\varphi}^* \in \arg\min_{\boldsymbol{\varphi}\in\Phi} \mathcal{L}^{(\boldsymbol{\varphi})}(\boldsymbol{\theta}^*, \boldsymbol{\varphi})$$

- In games we want to **converge** to the Saddle Point.
- Different from **single** objective **minimization** where we want to avoid saddle points.
- ~~Saddle point~~ -> **Zero-sum game (or Minmax)**

Variational Inequality Problem (VIP)

Variational Inequality Problem

- Based on **stationary conditions.**
- Relates to vast literature with standard algorithms.

Nash-Equilibrium: $\begin{cases} \theta^* = \arg\min_{\theta} L_\theta(\theta, \phi^*) \\ \phi^* = \arg\min_{\phi} L_\phi(\theta^*, \phi) \end{cases}$ ⟵ No player can improve its cost

Stationary Conditions: $\begin{cases} \nabla_\theta L_\theta(\theta^*, \phi^*)^T(\theta - \theta^*) \geq 0 \\ \nabla_\phi L_\phi(\theta^*, \phi^*)^T(\phi - \phi^*) \geq 0 \end{cases}$ $\forall (\theta, \phi) \in \Theta \times \Phi$

can be **constraint sets**.

Variational Inequality Problems

Nash-Equilibrium:

$$\begin{cases} \theta^* = \arg\min_{\theta} L_\theta(\theta, \phi^*) \\ \phi^* = \arg\min_{\phi} L_\phi(\theta^*, \phi) \end{cases}$$

(under convexity assumption)

$\Longleftrightarrow$

Stationary Conditions:

$$\begin{cases} \nabla_\theta L_\theta(\theta^*, \phi^*)^T(\theta - \theta^*) \geq 0 \\ \nabla_\phi L_\phi(\theta^*, \phi^*)^T(\phi - \phi^*) \geq 0 \end{cases} \quad \forall(\theta, \phi) \in \Theta \times \Phi$$

Same problem but **different** perspective.

Joint Minimization vs. **Stationary point**

Variational Inequality Problem

Stationary Conditions: $\begin{cases} \nabla_\theta L_\theta(\theta^*, \phi^*)^T(\theta - \theta^*) \geq 0 \\ \nabla_\phi L_\phi(\theta^*, \phi^*)^T(\phi - \phi^*) \geq 0 \end{cases} \quad \forall (\theta, \phi) \in \Theta \times \Phi$

Can be written as: $F(\omega) = \begin{pmatrix} \nabla_\theta L_\theta(\omega) \\ \nabla_\phi L_\phi(\omega) \end{pmatrix}$

$\omega = (\theta, \phi)$

$$F(\omega^*)^T(\omega - \omega^*) \geq 0 \quad \forall \omega \in \Omega$$

ω^* solves the **Variational Inequality**

Variational Inequality Problem

Stationary Conditions: $$F(\omega^*)^T(\omega - \omega^*) \geq 0 \quad \forall \omega \in \Omega$$

Unconstrained (or optimum in the interior):

$$\|\nabla_{\boldsymbol{\theta}} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)\| = \|\nabla_{\boldsymbol{\varphi}} \mathcal{L}^{(\boldsymbol{\varphi})}(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)\| = 0\,.$$

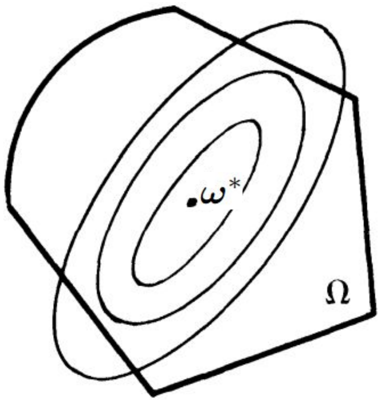


Figure from [Dunn 1979]

Variational Inequality Problem

Stationary Conditions: $$F(\omega^*)^T(\omega - \omega^*) \geq 0 \quad \forall \omega \in \Omega$$

Unconstrained (or ω* in the interior):

$$\|\nabla_{\boldsymbol{\theta}} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)\| = \|\nabla_{\boldsymbol{\varphi}} \mathcal{L}^{(\boldsymbol{\varphi})}(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)\| = 0\,.$$

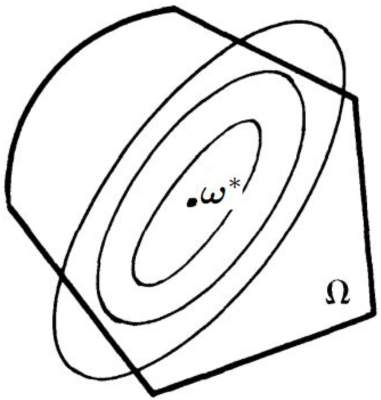


Figure from [Dunn 1979]

Constrained and ω* on the boundary:

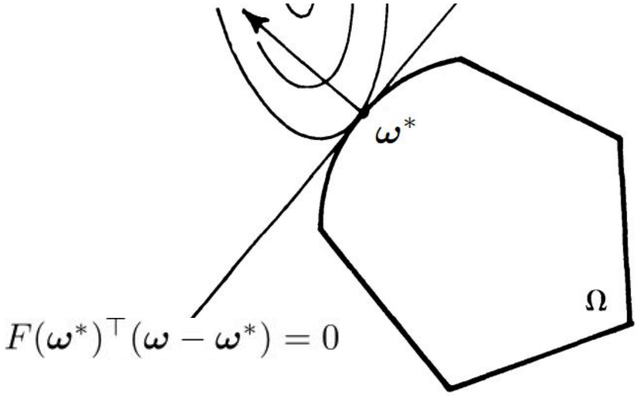


Figure from [Dunn 1979]

Gauthier Gidel,
Predoc III , November 28, 2018

Techniques to optimize VIP (Batch setting)

Standard Algorithms from Variational Inequality

Method 1: **Averaging**

- Converge even for *"cycling behavior"*.
- Easy to implement. (out of the training loop)
- Can be combined with any method.

$$\bar{\omega}_T \stackrel{\text{def}}{=} \frac{\sum_{t=0}^{T-1} \rho_t \omega_t}{S_T}, \quad S_T \stackrel{\text{def}}{=} \sum_{t=0}^{T-1} \rho_t \,.$$

Averaging schemes can be efficiently implemented in an **online** fashion:

$$\bar{\omega}_t = (1 - \tilde{\rho}_t)\bar{\omega}_{t-1} + \tilde{\rho}_t \omega_t \quad \text{where} \quad 0 \leq \tilde{\rho}_t \leq 1 \,.$$

Standard Algorithms from Variational Inequality

Method 1: **Averaging**

- Converge even for *"cycling behavior"*.
- Easy to implement. (out of the training loop)
- Can be combined with any method.

General Online averaging: $\bar{\omega}_t = (1 - \tilde{\rho}_t)\bar{\omega}_{t-1} + \tilde{\rho}_t \omega_t \quad \text{where} \quad 0 \leq \tilde{\rho}_t \leq 1\,.$

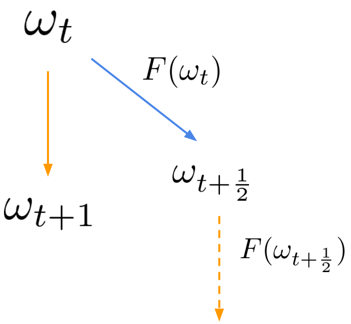
<u>Example 1:</u> **Uniform** averaging $\tilde{\rho}_t = \frac{1}{t}\,,\ t \geq 0 : \quad \bar{\omega}_T = \frac{1}{T}\sum_{k=0}^{T-1} \omega_t$

<u>Example 2:</u>
Exponential moving averaging (EMA) $\tilde{\rho}_t = 1 - \beta < 1\,,\ t \geq 0 : \quad \bar{\omega}_T = (1-\beta)\sum_{t=1}^{T} \beta^{T-t}\omega_t + \beta^T \omega_0$

Standard Algorithms from Variational Inequality

Method 2: **Extragradient**

- Step 1: $\omega_{t+\frac{1}{2}} = \omega_t - \gamma_t F(\omega_t)$
- Step 2: $\omega_{t+1} = \omega_t - \gamma_t F(\omega_{t+\frac{1}{2}})$



- Standard in the literature.
- Does not require *averaging.*
- *Theoretically* and *empirically* **faster.**

Intuition:

1. *<u>Game prespective:</u> Look one step in the future and anticipate next move of adversary.*

Frank-Wolfe Algorithm for Saddle Point Problems

Based on an AISTATS paper [Gidel et al. 2017].
Joint work with Tony Jebara and Simon Lacoste-Julien

Saddle point problems

$$\min_{\boldsymbol{\theta} \in \Theta} \max_{\boldsymbol{\phi} \in \Phi} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$$

Smooth, **differentiable** cost function,

- → Compact **constraints** sets.
- → Looking for stationary (fixed) points
- → Gradient ~~descent~~ **method.**

Saddle point problems

$$\min_{\boldsymbol{\theta} \in \Theta} \max_{\boldsymbol{\phi} \in \Phi} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$$

Smooth, **differentiable** cost function,

→ Compact **constraints** sets. → **Need to project ?**

→ Looking for stationary (fixed) points

→ Gradient ~~descent~~ **method.**

Projection-free Method

(Extra-)Gradient method:

- Require **Projection**
- Each projection is a **quadratic** problem

$$P_\Omega[\boldsymbol{\omega}] := \min_{\boldsymbol{\omega}' \in \Omega} \|\boldsymbol{\omega} - \boldsymbol{\omega}'\|_2^2$$

- Might be too expensive if the constraints set is **structured.**
- May use instead **projection-free** methods.
- Frank-Wolfe is projection-free.
- It only requires to solve **linear** problem.

$$\mathrm{LMO}[\boldsymbol{v}] := \min_{\boldsymbol{\omega} \in \Omega} \boldsymbol{\omega}^\top \boldsymbol{v}$$

<u>Projection may be challenging.</u>

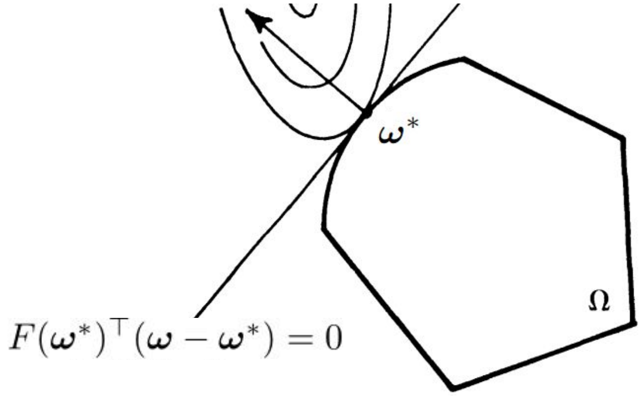


Figure from [Dunn 1979]

Projection-free Method

(Extra-)Gradient method:

- Require **Projection**
- Each projection is a **quadratic** problem

$$P_\Omega[\boldsymbol{\omega}] := \min_{\boldsymbol{\omega}' \in \Omega} \|\boldsymbol{\omega} - \boldsymbol{\omega}'\|_2$$

- Might be too expensive if the constraints set is **structured.**
- May use instead **projection-free** methods.
- Frank-Wolfe is projection-free.
- It only requires to solve **linear** problem.

$$\mathrm{LMO}[\boldsymbol{v}] := \min_{\boldsymbol{\omega} \in \Omega} \boldsymbol{\omega}^\top \boldsymbol{v}$$

Example of problem with expensive projection:

The ***structured SVM:***

$$\min_{\omega \in \mathbb{R}^d} \lambda\Omega(\omega) + \frac{1}{n}\sum_{i=1}^{n} \underbrace{\max_{y \in \mathcal{Y}_i} \left(L_i(y) - \langle \omega, \phi_i(y)\rangle\right)}_{\text{structured hinge loss}}$$

Regularization: penalized $\rightarrow$ constrained.

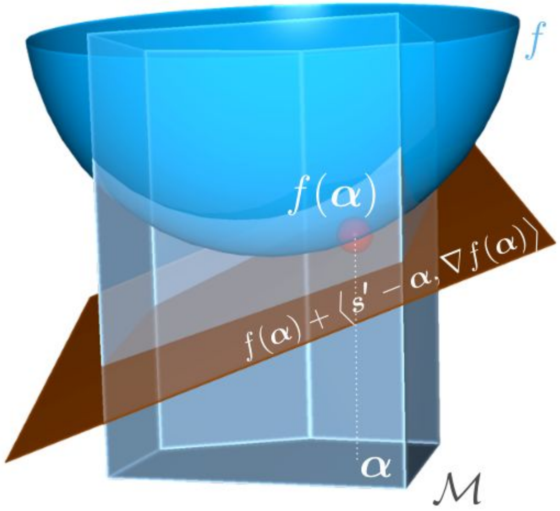
$$\min_{\Omega(\omega)\leq\beta} \max_{\alpha \in \Delta(|\mathcal{Y}|)} b^T\alpha - \omega^T M\alpha$$

Gauthier Gidel,
Predoc III , November 28, 2018

Projection-free Method

Algorithm Frank-Wolfe algorithm

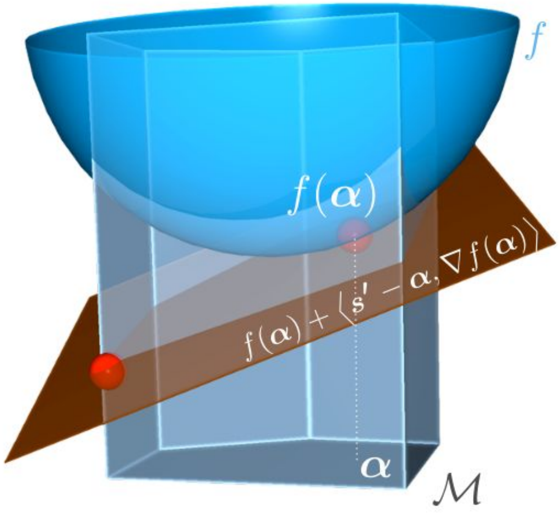
Let $\boldsymbol{x}^{(0)} \in \mathcal{X}$
for $t = 0 \ldots T$ **do**
  Compute $\boldsymbol{r}^{(t)} = \nabla f(\boldsymbol{x}^{(t)})$
  Compute $\boldsymbol{s}^{(t)} \in \operatorname*{argmin}_{\boldsymbol{s} \in \mathcal{X}} \left\langle \boldsymbol{s}, \boldsymbol{r}^{(t)} \right\rangle$
  Compute $g_t := \left\langle \boldsymbol{x}^{(t)} - \boldsymbol{s}^{(t)}, \boldsymbol{r}^{(t)} \right\rangle$
  if $g_t \leq \epsilon$ **then return** $\boldsymbol{x}^{(t)}$
  Let $\gamma = \frac{2}{2+t}$ (or do line-search)
  Update $\boldsymbol{x}^{(t+1)} := (1-\gamma)\boldsymbol{x}^{(t)} + \gamma \boldsymbol{s}^{(t)}$
end for



Projection-free Method

Algorithm Frank-Wolfe algorithm

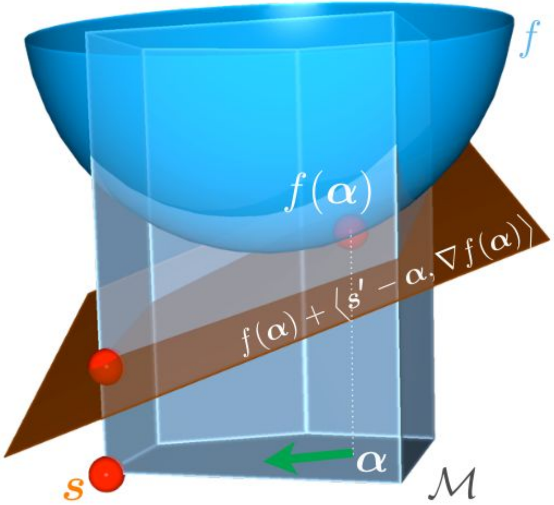
Let $\boldsymbol{x}^{(0)} \in \mathcal{X}$
for $t = 0 \dots T$ **do**
  Compute $\boldsymbol{r}^{(t)} = \nabla f(\boldsymbol{x}^{(t)})$
  Compute $\boldsymbol{s}^{(t)} \in \underset{\boldsymbol{s} \in \mathcal{X}}{\operatorname{argmin}} \left\langle \boldsymbol{s}, \boldsymbol{r}^{(t)} \right\rangle$
  Compute $g_t := \left\langle \boldsymbol{x}^{(t)} - \boldsymbol{s}^{(t)}, \boldsymbol{r}^{(t)} \right\rangle$
  if $g_t \leq \epsilon$ **then return** $\boldsymbol{x}^{(t)}$
  Let $\gamma = \frac{2}{2+t}$ (or do line-search)
  Update $\boldsymbol{x}^{(t+1)} := (1-\gamma)\boldsymbol{x}^{(t)} + \gamma \boldsymbol{s}^{(t)}$
end for



Projection-free Method

Algorithm Frank-Wolfe algorithm

Let $\boldsymbol{x}^{(0)} \in \mathcal{X}$
for $t = 0 \dots T$ **do**
  Compute $\boldsymbol{r}^{(t)} = \nabla f(\boldsymbol{x}^{(t)})$
  Compute $\boldsymbol{s}^{(t)} \in \underset{\boldsymbol{s} \in \mathcal{X}}{\operatorname{argmin}} \left\langle \boldsymbol{s}, \boldsymbol{r}^{(t)} \right\rangle$
  Compute $g_t := \left\langle \boldsymbol{x}^{(t)} - \boldsymbol{s}^{(t)}, \boldsymbol{r}^{(t)} \right\rangle$
  if $g_t \leq \epsilon$ **then return** $\boldsymbol{x}^{(t)}$
  Let $\gamma = \frac{2}{2+t}$ (or do line-search)
  Update $\boldsymbol{x}^{(t+1)} := (1-\gamma)\boldsymbol{x}^{(t)} + \gamma \boldsymbol{s}^{(t)}$
end for



Mila

Projection-free Method for Saddle Point

Algorithm Saddle point FW algorithm

Let $\boldsymbol{z}^{(0)} = (\boldsymbol{x}^{(0)}, \boldsymbol{y}^{(0)}) \in \mathcal{X} \times \mathcal{Y}$
for $t = 0 \dots T$ **do**
    Compute $\boldsymbol{r}^{(t)} := \begin{pmatrix} \nabla_x \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) \\ -\nabla_y \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) \end{pmatrix}$
    Compute $\boldsymbol{s}^{(t)} \in \underset{z \in \mathcal{X} \times \mathcal{Y}}{\operatorname{argmin}} \left\langle \boldsymbol{z}, \boldsymbol{r}^{(t)} \right\rangle$
    Compute $g_t := \left\langle \boldsymbol{z}^{(t)} - \boldsymbol{s}^{(t)}, \boldsymbol{r}^{(t)} \right\rangle$
    if $g_t \leq \epsilon$ **then return** $\boldsymbol{z}^{(t)}$
    Let $\gamma = \min\left(1, \frac{\nu}{C} g_t\right)$ **or** $\gamma = \frac{2}{2+t}$
    Update $\boldsymbol{z}^{(t+1)} := (1-\gamma)\boldsymbol{z}^{(t)} + \gamma \boldsymbol{s}^{(t)}$
end for

Theoretical Contributions

SP extension of FW with *away step*:

Convergence: $\boxed{\begin{array}{c}\textbf{\textit{Linear}} \text{ rate with } \textbf{\textit{adaptive}} \text{ step size.} \\ \textbf{\textit{Sublinear}} \text{ rate with } \textbf{\textit{universal}} \text{ step size.}\end{array}}$

- Similar hypothesis as AFW for linear convergence:
  1. Strong convexity and smoothness of the function.
  2. $\mathcal{X}$ and $\mathcal{Y}$ polytopes.
- Additional assumption on the bilinearity.

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = f(\boldsymbol{\theta}) + \boldsymbol{\theta}^\top M \boldsymbol{\phi} - g(\boldsymbol{\phi})$$

  $\|M\|$ smaller than the strong convexity constant.

- Proof use recent advances on AFW.

  Partially answering a **30 years old conjecture** .[Hammond 1984]

Negative Momentum for Improved Game Dynamics

Based on an AISTATS submission [Gidel et al. 2018b].

Joint work with Reyhane Askari Hemmat, Mohammad Pezeshki, Rémi Le Priol, Gabriel Huang, Simon Lacoste-Julien and Ioannis Mitliagkas

Two-player Games

Nash Equilibrium

$$\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta} \in \boldsymbol{\theta}} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}, \boldsymbol{\varphi}^*)$$

$$\boldsymbol{\varphi}^* \in \arg\min_{\boldsymbol{\varphi} \in \boldsymbol{\varphi}} \mathcal{L}^{(\boldsymbol{\varphi})}(\boldsymbol{\theta}^*, \boldsymbol{\varphi})$$

Smooth, differentiable L
→ Looking for local Nash equil.

→ Gradient method:
- → **Simultaneous**
- → **Alternating**

Two-player Games

Simultaneous Updates:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}_t, \phi_t)$$
$$\phi_{t+1} = \phi_t - \eta \nabla_{\phi} \mathcal{L}^{(\phi)}(\boldsymbol{\theta}_t, \phi_t)$$

Alternating Updates:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}_t, \phi_t)$$
$$\phi_{t+1} = \phi_t - \eta \nabla_{\phi} \mathcal{L}^{(\phi)}(\boldsymbol{\theta}_{t+1}, \phi_t)$$

First contribution: Bilinear game

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\varphi}} \; \boldsymbol{\theta}^\top \boldsymbol{A} \boldsymbol{\varphi}$$

Method	β	Bounded	Converges
Simultaneous	$\beta \in \mathbb{R}$	✗	✗
Alternated	>0	✗	✗
	0	✓	✗
	<0	✓	✓

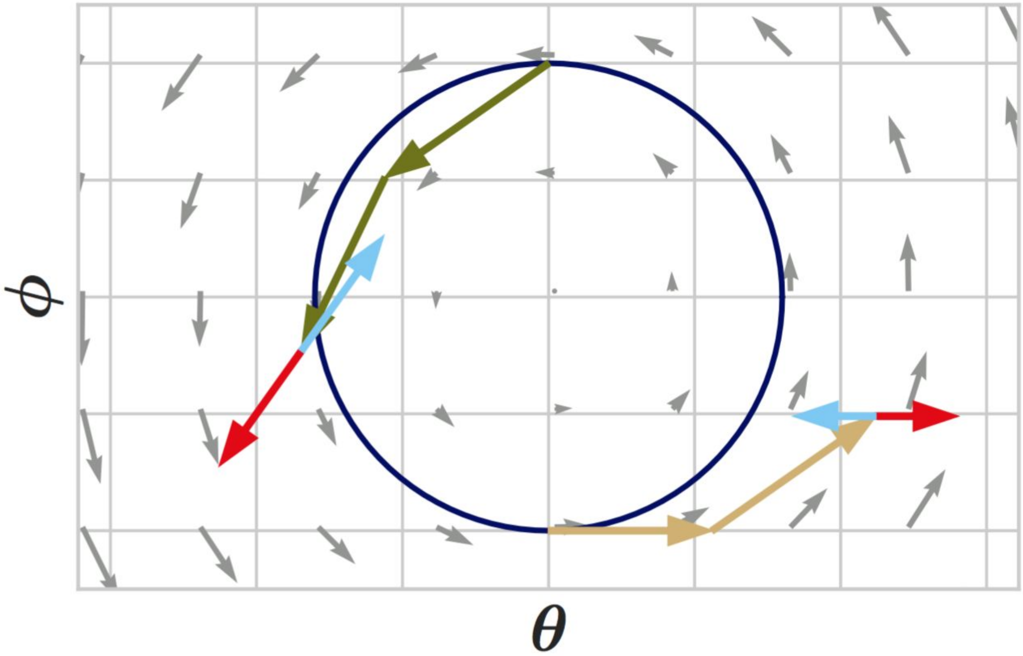
“Proof by picture”

Gradient descent

- → Simultaneous
- → Alternating

Momentum

- → Positive
- → Negative



Second contribution: Game dynamics

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}_t, \phi_t)$$
$$\phi_{t+1} = \phi_t - \eta \nabla_{\phi} \mathcal{L}^{(\phi)}(\boldsymbol{\theta}_t, \phi_t)$$

$$\boldsymbol{v}(\boldsymbol{\varphi}, \boldsymbol{\theta}) := \begin{bmatrix} \nabla_{\boldsymbol{\varphi}} \mathcal{L}^{(\boldsymbol{\varphi})}(\boldsymbol{\varphi}, \boldsymbol{\theta}) \\ \nabla_{\boldsymbol{\theta}} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\varphi}, \boldsymbol{\theta}) \end{bmatrix}$$

$$F_\eta(\boldsymbol{\varphi}, \boldsymbol{\theta}) \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{\varphi} & \boldsymbol{\theta} \end{bmatrix}^\top - \eta\, \boldsymbol{v}(\boldsymbol{\varphi}, \boldsymbol{\theta})$$

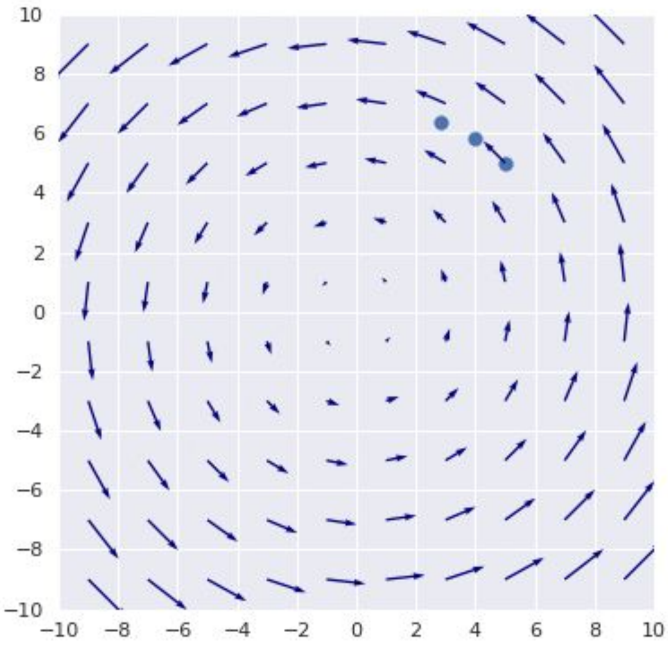
Game dynamics under gradient descent

$$F_\eta(\boldsymbol{\varphi},\boldsymbol{\theta}) \overset{\text{def}}{=} [\boldsymbol{\varphi} \quad \boldsymbol{\theta}]^\top - \eta\, \boldsymbol{v}(\boldsymbol{\varphi},\boldsymbol{\theta})$$

Jacobian is non-symmetric, with complex eigenvalues → Rotations in decision space

Momentum can manipulate the eigenvalues of the Jacobian.

Can momentum help/hurt??



Spoiler

Positive momentum can be bad for adversarial games

Practice that was very common when GANs were first invented.

→ Recent work reduced the momentum parameter.

→ Not an accident

Momentum on games

Recall Polyak's momentum (on top of simultaneous grad. desc.):

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta \boldsymbol{v}(\boldsymbol{x}_t) + \beta(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}), \quad \boldsymbol{x}_t = (\boldsymbol{\theta}_t, \boldsymbol{\phi}_t)$$

Fixed point operator requires a **state augmentation**:
(because we need previous iterate)

$$F_{\eta,\beta}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) := \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{0}_n \\ \boldsymbol{I}_n & \boldsymbol{0}_n \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{x}_{t-1} \end{bmatrix} - \eta \begin{bmatrix} \boldsymbol{v}(\boldsymbol{x}_t) \\ \boldsymbol{0}_n \end{bmatrix} + \beta \begin{bmatrix} \boldsymbol{I}_n & -\boldsymbol{I}_n \\ \boldsymbol{0}_n & \boldsymbol{0}_n \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{x}_{t-1} \end{bmatrix}$$

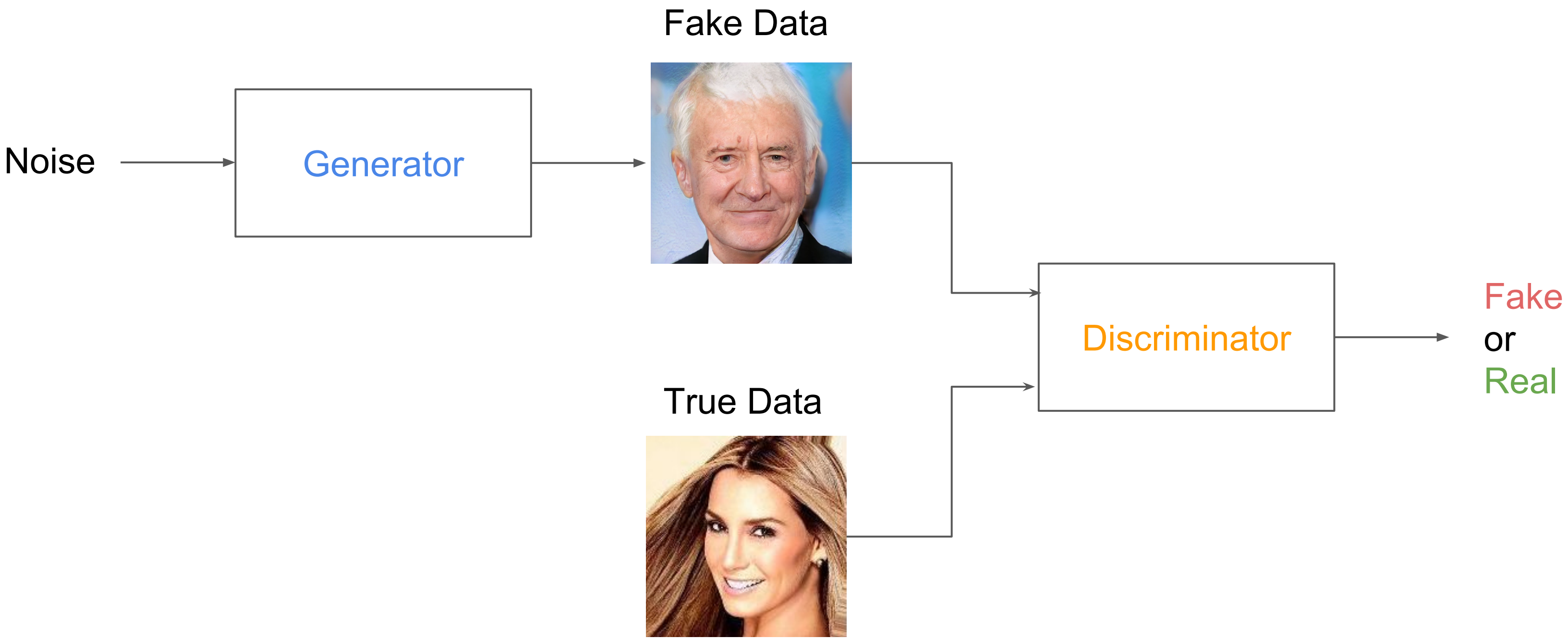
A Variational Inequality Perspective on GANs

Based on an ICLR submission [Gidel et al. 2018a].
Joint work with Hugo Berard, Gaëtan Vignoud, Pascal Vincent, Simon Lacoste-Julien

Quick recap on Generative Adversarial Networks (GANs)
(and two-player games)

Generative Adversarial Networks (GANs)

[Goodfellow et al. NIPS 2014]



Generative Adversarial Networks (GANs)

[Goodfelow et al. NIPS 2014]

$$\min_{\theta} \underbrace{\max_{\phi} \mathbb{E}_{x\sim p_{\mathcal{D}}}[\log(\overset{\text{Discriminator}}{D_\phi}(x))] + \mathbb{E}_{z\sim p_{\mathcal{Z}}}[\log(1 - D_\phi(\overset{\text{Generator}}{G_\theta}(z)))]}$$

If D is non-parametric: $\quad L(\theta) = \mathrm{JSD}(p_{\mathcal{D}}||p_\theta)$

Non-saturating GAN: “much stronger gradient in early learning”

$$\overbrace{\min_{\theta} -\mathbb{E}_{z\sim p_{\mathcal{Z}}}[\log(D_\phi(G_\theta(z)))]}^{\text{Loss of Generator}} \qquad \overbrace{\max_{\phi} \mathbb{E}_{x\sim p_{\mathcal{D}}}[\log(D_\phi(x))] + \mathbb{E}_{z\sim p_{\mathcal{Z}}}[\log(1 - D_\phi(G_\theta(z)))]}^{\text{Loss of Discriminator}}$$

Two-player Games

Player 1 Player 2

$$\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}\in\Theta} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}, \boldsymbol{\varphi}^*) \quad \text{and} \quad \boldsymbol{\varphi}^* \in \arg\min_{\boldsymbol{\varphi}\in\Phi} \mathcal{L}^{(\boldsymbol{\varphi})}(\boldsymbol{\theta}^*, \boldsymbol{\varphi})$$

Non zero-sum game if we **do not** have: $\mathcal{L}^{(\boldsymbol{\theta})} = -\mathcal{L}^{(\boldsymbol{\varphi})}$

Example: Non-saturating GAN: [Goodfellow et al. 2014]

Loss of Generator

$$\min_{\theta} -\mathbb{E}_{z\sim p_{\mathcal{Z}}}[\log(D_\phi(G_\theta(z)))]$$

Loss of Discriminator

$$\max_{\phi} \mathbb{E}_{x\sim p_{\mathcal{D}}}[\log(D_\phi(x))] + \mathbb{E}_{z\sim p_{\mathcal{Z}}}[\log(1 - D_\phi(G_\theta(z)))]$$

GANs as a Variational Inequality

Takeaways:

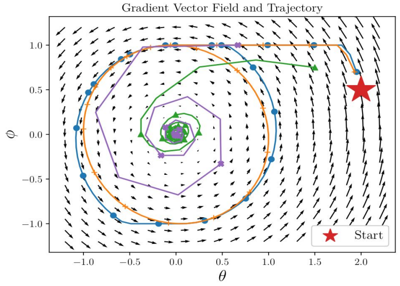
- GAN can be formulated as a **Variational Inequality.**
- Encompass most of GANs formulations.
- **Standard algorithms** from Variational Inequality can be used for GANs.
- **Theoretical Guarantees** (for convex and *stochastic* cost functions).

$$\begin{cases} \theta^* = \arg\min_\theta L_\theta(\theta, \phi^*) \\ \phi^* = \arg\min_\phi L_\phi(\theta^*, \phi) \end{cases}$$

$$\downarrow$$

$$F(\omega^*)^T(\omega - \omega^*) \geq 0 \quad \forall \omega \in \Omega$$

$$\downarrow$$



Standard Algorithms from Variational Inequality

Method 1: **Averaging**

- Converge even for *"cycling behavior"*.
- Easy to implement. (out of the training loop)
- Can be combined with any method.

General Online averaging: $\bar{\boldsymbol{\omega}}_t = (1 - \tilde{\rho}_t)\bar{\boldsymbol{\omega}}_{t-1} + \tilde{\rho}_t \boldsymbol{\omega}_t \quad \text{where} \quad 0 \leq \tilde{\rho}_t \leq 1\,.$

Example 1: Uniform averaging $\tilde{\rho}_t = \frac{1}{t}\,,\ t \geq 0 : \quad \bar{\boldsymbol{\omega}}_T = \frac{1}{T}\sum_{k=0}^{T-1} \boldsymbol{\omega}_t$

Example 2:
Exponential moving averaging (EMA) $\tilde{\rho}_t = 1 - \beta < 1\,,\ t \geq 0\ : \quad \bar{\omega}_T = (1-\beta)\sum_{t=1}^{T} \beta^{T-t}\omega_t + \beta^T \omega_0$

Standard Algorithms from Variational Inequality

Method 1: **Averaging**

Simple Minmax problem: $$\min_{\theta \in \mathbb{R}} \max_{\phi \in \mathbb{R}} \theta \cdot \phi \quad \Rightarrow \quad (\theta^*, \phi^*) = (0, 0)\,.$$

Simultaneous update: $$\begin{cases} \theta_{t+1} = \theta_t - \eta \phi_t \\ \phi_{t+1} = \phi_t + \eta \theta_t \end{cases},$$

Alternated update: $$\begin{cases} \theta_{t+1} = \theta_t - \eta \phi_t \\ \phi_{t+1} = \phi_t + \eta \theta_{t+1} \end{cases}$$

Standard Algorithms from Variational Inequality

Method 1: **Averaging**

Simple Minmax problem: $\min_{\theta \in \mathbb{R}} \max_{\phi \in \mathbb{R}} \; \theta \cdot \phi \quad \Rightarrow \quad (\theta^*, \phi^*) = (0, 0)\,.$

Simultaneous update: $\begin{cases} \theta_{t+1} = \theta_t - \eta \phi_t \\ \phi_{t+1} = \phi_t + \eta \theta_t \end{cases},$ **Alternated update:** $\begin{cases} \theta_{t+1} = \theta_t - \eta \phi_t \\ \phi_{t+1} = \phi_t + \eta \theta_{t+1} \end{cases}$

$$(\bar\theta_T, \bar\phi_T) := \frac{1}{T} \sum_{k=0}^{T-1} (\theta_t, \phi_t) \to \infty$$

$$(\theta_T, \phi_T) \to \infty$$

$$0 < m \le \|\theta_T, \phi_T\| \le M$$

$$(\bar\theta_T, \bar\phi_T) \to (0, 0)$$

Standard Algorithms from Variational Inequality

Method 1: **Averaging**

$$\min_{\theta \in \mathbb{R}} \max_{\phi \in \mathbb{R}} \quad \theta \cdot \phi \quad \Longrightarrow \quad (\theta^*, \phi^*) = (0, 0) \,.$$

Simultaneous update: $\begin{cases} \theta_{t+1} = \theta_t - \eta \phi_t \\ \phi_{t+1} = \phi_t + \eta \theta_t \end{cases}$,

$$(\bar\theta_T, \bar\phi_T) := \frac{1}{T} \sum_{k=0}^{T-1} (\theta_t, \phi_t) \to \infty \qquad (\theta_T, \phi_T) \to \infty$$

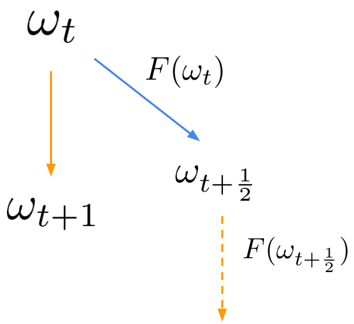
Alternated update: $\begin{cases} \theta_{t+1} = \theta_t - \eta \phi_t \\ \phi_{t+1} = \phi_t + \eta \theta_{t+1} \end{cases}$

$$0 < m \leq \|\theta_T, \phi_T\| \leq M \qquad (\bar\theta_T, \bar\phi_T) \to (0, 0)$$

Standard Algorithms from Variational Inequality

Method 2: **Extragradient**

- Step 1: $\omega_{t+\frac{1}{2}} = \omega_t - \gamma_t F(\omega_t)$
- Step 2: $\omega_{t+1} = \omega_t - \gamma_t F(\omega_{t+\frac{1}{2}})$



- Standard in the literature.
- Does not require *averaging.*
- *Theoretically* and *empirically* **faster.**

Intuition:

1. *<u>Game prespective:</u> Look one step in the future and anticipate next move of adversary.*
2. *Euler's method: Extrapolation is close to an **implicit** method because* $\omega_{t+1/2} \approx \omega_{t+1}$

$$\omega_{t+1} - \omega_{t+1/2} = O(\gamma_t^2)$$

Standard Algorithms from Variational Inequality

Method 2: **Extragradient**

Intuition: *Extrapolation is close to an* ***implicit*** *method because* $\omega_{t+1/2} \approx \omega_{t+1}$

$$\text{Implicit step:} \quad \omega_{t+1} = \omega_t - \eta F(\omega_{t+1})$$

Unknown:
Require to solve a
non-linear system

Standard Algorithms from Variational Inequality

Method 2: **Extragradient**

Intuition: *Extrapolation is close to an* ***implicit*** *method*

$$\min_{\theta \in \mathbb{R}} \max_{\phi \in \mathbb{R}} \ \theta \cdot \phi \qquad \text{and} \qquad (\theta^*, \phi^*) = (0, 0)\,.$$

$$\text{Implicit:} \begin{cases} \theta_{t+1} = \theta_t - \eta \phi_{t+1} \\ \phi_{t+1} = \phi_t + \eta \theta_{t+1} \end{cases}, \qquad \text{Extrapolation:} \begin{cases} \theta_{t+1} = \theta_t - \eta(\phi_t + \eta \theta_t) \\ \phi_{t+1} = \phi_t + \eta(\theta_t - \eta \phi_t) \end{cases}. \qquad (\,*\,)$$

Proposition 2. *The squared norm of the iterates $N_t \overset{def}{=} \theta_t^2 + \phi_t^2$, where the update rule of θ_t and ϕ_t are defined in (*), decreases geometrically for any $\eta < 1$ as,*

$$\text{Implicit:} \ N_{t+1} = (1 - \eta^2 + \eta^4 + \mathcal{O}(\eta^6))N_t\,, \qquad \text{Extrapolation:} \ N_{t+1} = (1 - \eta^2 + \eta^4)N_t\,.$$

almost the same

Mila Université de Montréal McGill

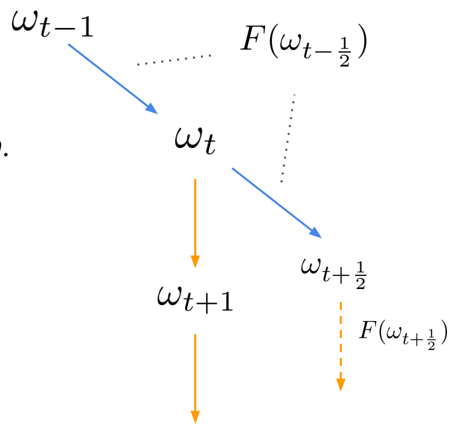
Gauthier Gidel,
Predoc III , November 28, 2018

Extrapolation from the past: Re-using the gradients

Problem: Extragradient requires to compute **two** gradients at each step.

Solution: **Extrapolation from the past** ⟵ **Re-use** gradient.

- Step 1: $\omega_{t+\frac{1}{2}} = \omega_t - \gamma_t F(\omega_{t-\frac{1}{2}})$ ⟵ ***Re-use*** *from previous iteration.*
- Step 2: $\omega_{t+1} = \omega_t - \gamma_t F(\omega_{t+\frac{1}{2}})$ ⟵ (*same as* ***extragradient***).

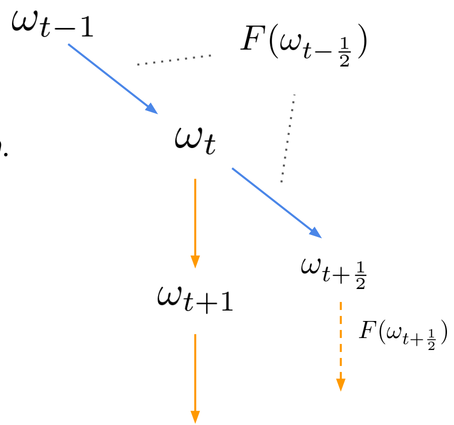


Extrapolation from the past: Re-using the gradients

Problem: Extragradient requires to compute **two** gradients at each step.

Solution: **Extrapolation from the past** ← **Re-use** gradient.

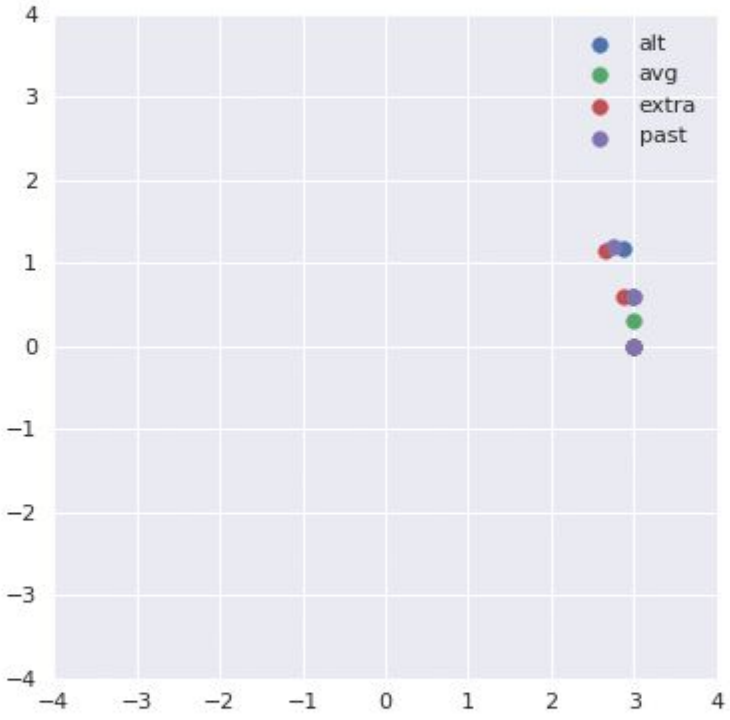
- Step 1: $\omega_{t+\frac{1}{2}} = \omega_t - \gamma_t F(\omega_{t-\frac{1}{2}})$ ← ***Re-use*** *from previous iteration.*
- Step 2: $\omega_{t+1} = \omega_t - \gamma_t F(\omega_{t+\frac{1}{2}})$ ← (*same as* ***extragradient***).



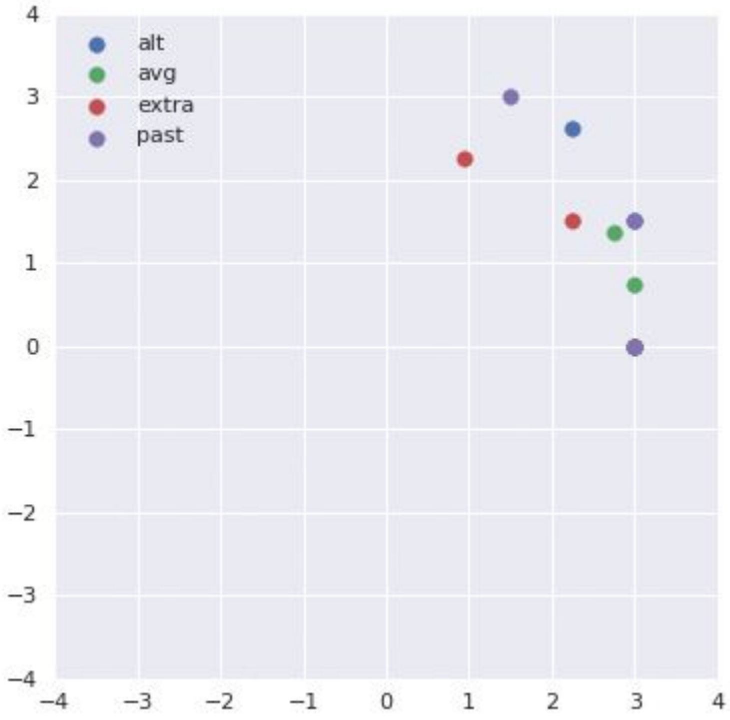
New Method !!!

Related to [Daskalakis et al., 2018]

step-size = 0.2



step-size = 0.5

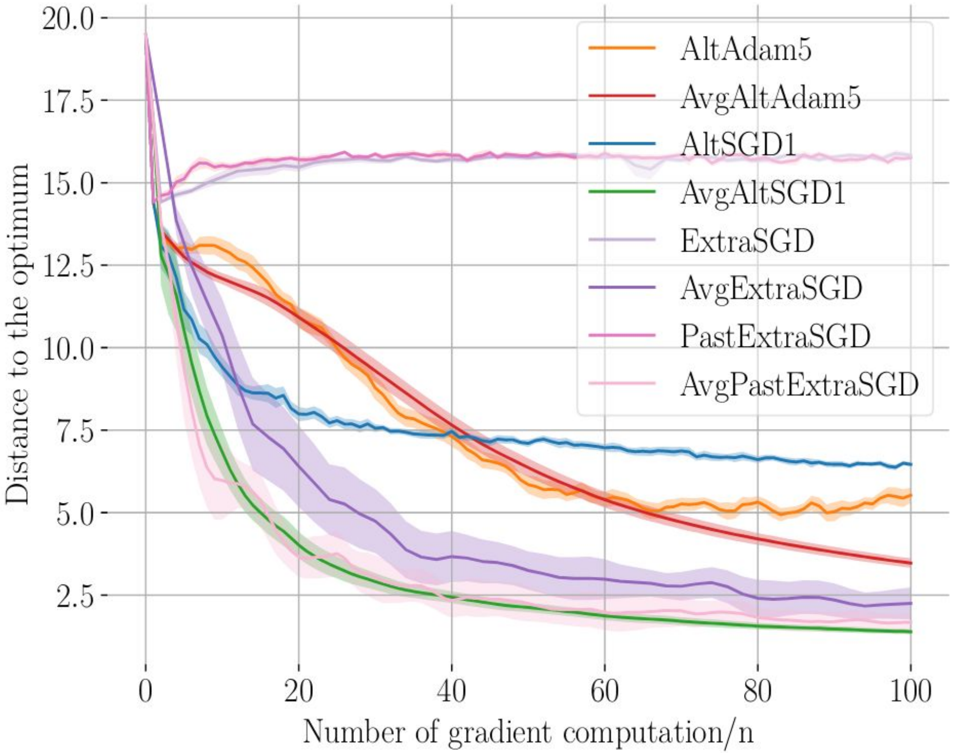


Experimental Results

Experimental Results

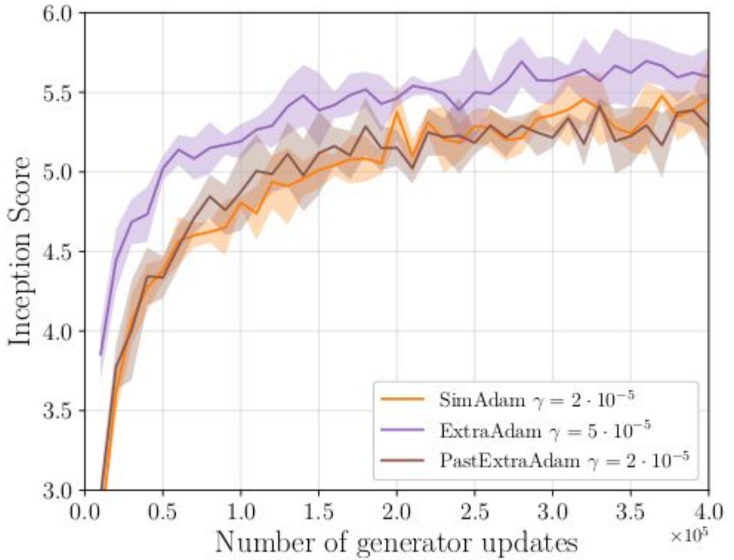
Bilinear *Stochastic* Objective:

$$\frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{x}^\top \boldsymbol{M}^{(i)}\boldsymbol{y} + \boldsymbol{x}^\top \boldsymbol{a}^{(i)} + \boldsymbol{y}^\top \boldsymbol{b}^{(i)}\right).$$



Experimental Results: WGAN (DCGAN) on CIFAR10

Inception Score vs
nb of generator updates



Inception Score on CIFAR10

Model	WGAN		
Method	no averaging	uniform avg	EMA
SimAdam	*$6.05 \pm .12$*	$5.83 \pm .16$	$6.08 \pm .10$
AltAdam5	*$5.45 \pm .08$*	$5.72 \pm .06$	$5.49 \pm .05$
ExtraAdam	$\mathbf{6.38 \pm .09}$	$\mathbf{6.38 \pm .20}$	$\mathbf{6.37 \pm .08}$
PastExtraAdam	5.98 ± 0.15	6.07 ± 0.19	6.01 ± 0.11
OptimAdam	5.74 ± 0.10	5.80 ± 0.08	5.78 ± 0.05

Extragradient Methods

Averaging

Algorithm 4 Extra-Adam: proposed Adam with extrapolation step.

input: step-size η, decay rates for moment estimates β_1, β_2, access to the stochastic gradients $\nabla \ell_t(\cdot)$ and to the projection $P_\Omega[\cdot]$ onto the constraint set Ω, initial parameter $\boldsymbol{\omega}_0$, averaging scheme $(\rho_t)_{t \geq 1}$

for $t = 0 \ldots T-1$ **do**

Option 1: Standard extrapolation.

Sample new minibatch and compute stochastic gradient: $g_t \leftarrow \nabla \ell_t(\boldsymbol{\omega}_t)$

Option 2: Extrapolation from the past

Load previously saved stochastic gradient: $g_t = \nabla \ell_{t-1/2}(\boldsymbol{\omega}_{t-1/2})$

Update estimate of first moment for extrapolation: $m_{t-1/2} \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$

Update estimate of second moment for extrapolation: $v_{t-1/2} \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2$

Correct the bias for the moments: $\hat{m}_{t-1/2} \leftarrow m_{t-1/2}/(1-\beta_1^{2t-1})$, $\hat{v}_{t-1/2} \leftarrow v_{t-1/2}/(1-\beta_2^{2t-1})$

Perform *extrapolation* step from iterate at time t: $\boldsymbol{\omega}_{t-1/2} \leftarrow P_\Omega[\boldsymbol{\omega}_t - \eta \frac{m_{t-1/2}}{\sqrt{v_{t-1/2}}+\epsilon}]$

Extrapolation (Adam style)

Sample new minibatch and compute stochastic gradient: $g_{t+1/2} \leftarrow \nabla \ell_{t+1/2}(\boldsymbol{\omega}_{t+1/2})$

Update estimate of first moment: $m_t \leftarrow \beta_1 m_{t-1/2} + (1-\beta_1) g_{t+1/2}$

Update estimate of second moment: $v_t \leftarrow \beta_2 v_{t-1/2} + (1-\beta_2) g_{t+1/2}^2$

Compute bias corrected for first and second moment: $\hat{m}_t \leftarrow m_t/(1-\beta_1^{2t})$, $\hat{v}_t \leftarrow v_t/(1-\beta_2^{2t})$

Perform *update* step from the iterate at time t: $\boldsymbol{\omega}_{t+1} \leftarrow P_\Omega[\boldsymbol{\omega}_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon}]$

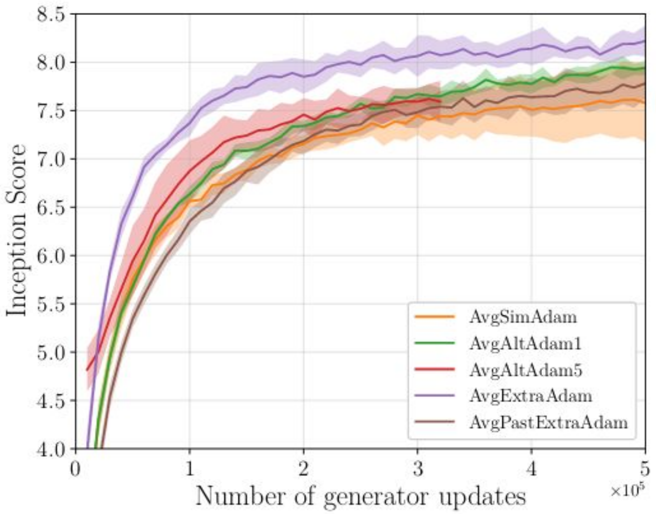
Update (Adam style)

end for

Output: $\boldsymbol{\omega}_{T-1/2}$, $\boldsymbol{\omega}_T$ or $\bar{\boldsymbol{\omega}}_T = \sum_{t=0}^{T-1} \rho_{t+1} \boldsymbol{\omega}_{t+1/2} / \sum_{t=0}^{T-1} \rho_{t+1}$ (see (8) for online averaging)

Experimental Results: WGAN-GP (ResNet) on CIFAR10

Inception Score vs
Number of



Model	WGAN-GP (ResNet)	
Method	no averaging	uniform avg
SimAdam	$7.54 \pm .21$	$7.74 \pm .27$
AltAdam5	$7.20 \pm .06$	$7.67 \pm .15$
ExtraAdam	$7.79 \pm .09$	$\mathbf{8.26 \pm .12}$
PastExtraAdam	$7.71 \pm .12$	$7.84 \pm .18$
OptimAdam	$7.80 \pm .07$	$\mathbf{7.99 \pm .12}$

Extragradient Methods

Averaging

Mila Université de Montréal McGill

Gauthier Gidel,
Predoc III , November 28, 2018

Conclusion

- Training of adversarial formulations has been a recurrent issue in modern ML.
- Impact of **non-convexity** and **stochasticity** are less understood than in the single objective minimization.
- A better understanding of this framework is **key** to design new optimization algorithms.
- We provided **tools** to better understand saddle point problem, multi-player games and more generally variational inequalities.
- However, we **just scratched the surface** .

Thank you !

Hugo Berard

Reyhane Askari Hemmat

Gaëtan Vignoud

Gabriel Huang

Rémi Le priol

Mohammad Pezeshki

Ioannis Mitliagkas

Simon Lacoste-Julien

Pascal Vincent

Tony Jebara

Bibliography

- Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein gan." in ICML 2017.

- Dunn, Joseph C. "Rates of convergence for conditional gradient algorithms near singular and nonsingular extremals." *SIAM Journal on Control and Optimization"* 1979

- Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." In *Advances in neural information processing systems* 2014.

- Gidel, Gauthier, Tony Jebara, and Simon Lacoste-Julien. "Frank-wolfe algorithms for saddle point problems." in *AISTATS* 2017.

- Gidel, Gauthier, Hugo Berard, Gaëtan Vignoud, Pascal Vincent, and Simon Lacoste-Julien. "A Variational Inequality Perspective on Generative Adversarial Nets." *arXiv preprint arXiv:1802.10551* (2018).

- Gidel, Gauthier, Reyhane Askari Hemmat, Mohammad Pezeshki, Rémi Le priol, Gabriel Huang, Simon Lacoste-Julien, and Ioannis Mitliagkas. "Negative momentum for improved game dynamics." *arXiv preprint arXiv:1807.04740* (2018).

- Hammond, Janice H. "Solving asymmetric variational inequality problems and systems of equations with generalized nonlinear programming algorithms." PhD diss., Massachusetts Institute of Technology, 1984.