A Variational Inequality Perspective on Generative Adversarial Networks

Gauthier Gidel*, 1, Hugo Berard*, 1,2, Gaëtan Vignoud¹, Pascal Vincent¹,2,3 and Simon Lacoste-Julien¹,3

*Equal contribution; ¹Mila, U. Montréal; ²Facebook AI Research; ³CIFAR fellow

Overview

- We survey the “variational inequality” framework.
- Encompasses all GAN training methods using gradients.
- Tapping into the mathematical programming literature, we counter some common misconceptions about the difficulties of saddle point optimization.

Contributions & Related Work

Contributions:
- Extend standard methods designed for variational inequalities to the training of GANs.
- Amongst others, we apply extrapolation and averaging to the stochastic gradient method (SGD) and Adam, to improve the training of GANs.
- We propose extrapolation from the past a cheaper variant of martingale.

Related work:
- Extragradient methods have been originally introduced by Korpelevich [5] and extended by Nesterov [7] to improve the training of GANs.
- Recently, Daskalakis et al. [2] proposed a method inspired from game theory related to extrapolation.
- Alternative to extragradient: negative momentum proposed by Gidel et al. [3].

GAs as a Variational Inequality

Unconstrained: point with zero gradient.
\[ \nabla L^G(\theta, \varphi) = 0. \]

Constrained: no feasible descent directions.
\[ \nabla L^G(\theta, \varphi) [\theta - \theta^*] \geq 0, \quad \forall \theta \in \Theta, \quad \nabla L^G(\theta, \varphi) [\varphi - \varphi^*] \geq 0, \quad \forall \varphi \in \Phi. \]

Variational Inequality Problem (VIP)

Defining \( \omega = (\theta, \varphi) \), \( \omega^* = (\theta^*, \varphi^*) \), \( \Omega = \Theta \times \Phi \), can be compactly formulated as:
\[ F(\omega^*) (\omega - \omega^*) \geq 0, \quad \forall \omega \in \Omega, \]
where \( F(\omega) \defeq \nabla L^G(\theta, \varphi) [\nabla L^G(\theta, \varphi)]^\top. \)

Takeaway:
- GAN can be formulated as a Variational Inequality.
- Encompasses most of GANs formulations.
- Standard algorithms from Variational Inequality can be applied to GANs.
- Theoretical Guarantees (for convex and stochastic cost functions).

Background

Two-player games

Two-player games Generalizes mini-max formulation:
\[ \theta^* = \arg \min_{\theta} \max_{\varphi} L^G(\theta, \varphi), \quad \varphi^* = \arg \min_{\varphi} \max_{\theta} L^D(\theta, \varphi), \]
\[ L^G(\theta, \varphi) = -\mathbb{E}_{x \sim \mathcal{D}} \log f_{\theta}(x), \quad L^D(\theta, \varphi) = -\mathbb{E}_{z \sim \mathcal{N}(0,1)} \log f_{\varphi}(x - \theta) \mathbb{1}_{x \in \mathcal{D}}. \]

WGAN [1] (zero-sum):
\[ \min_{\theta} \max_{\varphi} \mathbb{E}_{x \sim \mathcal{D}} [f_{\theta}(x)] - \mathbb{E}_{z \sim \mathcal{N}(0,1)} [f_{\varphi}(z)]. \]

The bilinear WGAN

The discriminator and the generator are linear:
\[ D_{\varphi}(x) = f_{\varphi}(x) = \varphi^\top x, \quad G_{\theta}(z) = \theta z \]
By replacing these expressions in the WGAN objective (1),
\[ \min_{\theta} \max_{\varphi} \mathbb{E}_{X \sim \mathcal{D}} [\varphi^\top \log \varphi - \varphi^\top \log \varphi]. \]

Algorithms for VIP

Averaging

\[ \omega_{t+1} = \frac{\omega_t + \theta_t}{2} \]
- Converges even for “cycling behavior”.
- Easy to implement. Can combine with any method.
- Can be implemented on an online fashion:
  - Uniform averaging:
    \[ \omega_{t+1} = \frac{\omega_t + \theta_t}{2} + \frac{1}{\beta} \omega^*. \]
  - EMA:
    \[ \omega_{t+1} = \omega_t + \frac{1 - \beta}{\beta} \omega^*. \]

Extragradient

\[ \omega_{t+1} = \omega_t - \gamma \nabla f(\omega_t) \quad \text{(extrap step)} \]
\[ \omega_{t+1} = \omega_t - \gamma \nabla f(\omega_{t+1}) \quad \text{(update step)} \]
Intuition: Look 1 step in the future and anticipate next move of adversary. Close to an implicit method.
- Does not require averaging.
- Theoretically and empirically faster.

Extrapolation from the past

Problem: Extragradient requires to compute two gradients at each step. (Twice as expensive !)
Solution: Re-use the previous gradient.
\[ \omega_{t+1} = \omega_t - \gamma \nabla f(\omega_{t+1}) \quad \text{(re-use from previous step)} \]
\[ \omega_{t+1} = \omega_t - \gamma \nabla f(\omega_{t+1}) \quad \text{(same as extragradient)} \]

SimGSD: Both parameters are updated simultaneously.
AltSGD: variant of SGD where \( \theta \) is updated before \( \theta \).
AvgSGD: return the average of SimSGD iterates.
ExtraSVD: SVD with an extrapolation step.
PastExtraSVD: SVD with extrapolation from the past.

Simple stochastic bilinear example

\[ \mathbb{E}_{z \sim \mathcal{N}(0,1)} (\varphi^\top (y + x^\top a + y^\top b^\top)) = \mathbb{E}_{z \sim \mathcal{N}(0,1)} (\varphi^\top y) = \mathbb{E}_{z \sim \mathcal{N}(0,1)} (\varphi^\top a) = \mathbb{E}_{z \sim \mathcal{N}(0,1)} (\varphi^\top b^\top) = 0. \]

Experiments

Model WGAN-GP (ResNet)

Method no avg uniform avg EMA
SimAdam 7.51 ± 0.17 7.63 ± 0.37 7.60 ± 0.17
AltAdam5 7.57 ± 0.62 8.01 ± 0.65 7.66 ± 0.03
ExtraAdam 7.90 ± 11.847 ± 10 8.13 ± 0.07
PastExtraAdam 8.74 ± 0.06 8.01 ± 0.09 7.99 ± 0.03
OptimAdam 7.98 ± 0.08 8.18 ± 0.09 8.10 ± 0.06

Frechet Inception Distance

Model WGAN-GP (ResNet)

Method no averaging uniform avg EMA
SimAdam 23.74 ± 2.79 26.29 ± 5.36 25.89 ± 2.51
AltAdam5 21.65 ± 0.66 19.91 ± 0.43 20.69 ± 0.37
ExtraAdam 19.42 ± 15 18.13 ± 15.67 ± 21
PastExtraAdam 19.95 ± 0.38 22.45 ± 9.73 18.75 ± 0.40
OptimAdam 18.88 ± 0.55 21.23 ± 1.19 19.61 ± 0.32

References