Optimization in ML: Beyond gradient descent

Anima Anandkumar

Caltech

Competitive Gradient Descent

Florian Schäfer

A

To appear in NeurIPS 2019

Caltech

Single Agent Optimization

$$\min_x f(x)$$

- Clear Objective:
  (“Good local”) minimum of f
- Reliable Algorithmic Baseline:
  (Stochastic) gradient descent

Competitive Optimization

- Two agents seek to choose their respective decision variables x and y

$$\min_x f(x, y), \qquad \min_y g(x, y)$$

- Each objective depends on both players!
- Goals may conflict (e.g. $f \approx -g$)

Applications in ML

- Constrained optimization/Lagrange multipliers
- Robust optimization
- Robust statistics
- Variational representation of e.g. f-divergences
- Generative Adversarial Networks (GANs)
- Reinforcement Learning
- Many more?

Strategic Equilibria

- Find Nash/strategic equilibrium $(\bar{x}, \bar{y})$ s.t.

$$\bar{x} = \operatorname{argmin}_x f(x, \bar{y}),$$
$$\bar{y} = \operatorname{argmin}_y g(\bar{x}, y)$$

- In the nonconvex case: find local Nash equilibrium $(\bar{x}, \bar{y})$ such that

$$\bar{x} = \operatorname{argmin}_x \bar{f}_{(\bar{x}, \bar{y})}(x, \bar{y}),$$
$$\bar{y} = \operatorname{argmin}_y \bar{g}_{(\bar{x}, \bar{y})}(\bar{x}, y),$$

for local approximations $\bar{f}_{(\bar{x}, \bar{y})}, \bar{g}_{(\bar{x}, \bar{y})}$.

Alternating Gradient Descent

- Simplest algorithm

$$x_{k+1} = x_k - \eta \nabla_x f$$
$$y_{k+1} = y_k - \eta \nabla_y g$$

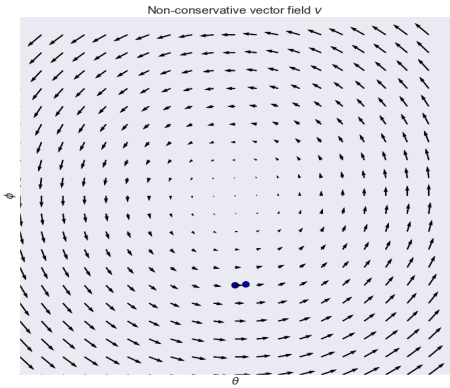
- Is it the competitive analogue of Gradient Descent?

Alternating Gradient Descent

- It fails even in the bilinear case:

$$f(x, y) = xy = -g(x, y)$$

- Instead of converging to $(0,0)$, the two players oscillate.
- Analogue of *"Rock! Paper! Scissor! Rock! Paper! …"*.
- Diverges for all finite stepsizes!



From https://www.inference.vc/my-notes-on-the-numerics-of-gans/ ©Ferenc Huszar

Alternating Gradient Descent

- Possible reason for mode collapse and oscillation in GANs.
- Lots of modifications proposed in the literature
  - *fictitious play* (Brown, 1951)
  - *predictive updates* (Yadav et al., 2017)
  - *follow the regularized leader* (Shalev-Shwartz and Singer, 2007; Grnarova et al., 2017)
  - *opponent learning awareness* (Foerster et al., 2018)
  - *optimism* (Rakhlin and Sridharan, 2013; Daskalakis et al., 2017; Mertikopoulos et al., 2019), cf also (Korpelevich, 1977)
  - *Consensus optimization* (Mescheder et al., 2017)
  - *Symplectic Gradient Adjustment* (Balduzzi et al., 2018; Letcher et al., 2019)
  - *...*

A polemic:

- Common point of view:
  The natural generalization of gradient descent is given by alternating gradient descent, which has poor convergence behavior and needs adjustments.
- Our point of view:
  The natural generalization of gradient descent is given by <u>another</u> algorithm, which has good convergence properties by default.

Recall Gradient Descent

- The gradient descent update is given as

$$x_{k+1} - x_k = \mathrm{argmin}_x \, f(x_k) + x^T \nabla_x f(x_k) + \frac{1}{2\eta} x^T x$$

- Optimizing local linear approximation, subject to a quadratic penalty

How to linearize a game?

- For two players:
  solution of local approximation, subject to quadratic penalty expressing the limited confidence of both players.

- What is the right way to linearize a game?

Linear or Multilinear?

- First attempt:
  Linear for one player → Linear for two players

$$x_{k+1} - x_k = \mathrm{argmin}_x \; f + x^T \nabla_x f + y^T \nabla_y f + \frac{1}{2\eta} x^T x$$

$$y_{k+1} - y_k = \mathrm{argmin}_y \; g + x^T \nabla_x g + y^T \nabla_y g + \frac{1}{2\eta} y^T y$$

- No interaction between the two players
  → We recover alternating gradient descent

Linear or Multilinear?

- Second attempt:
  Linear for one player → Bilinear for two players

$$x_{k+1} - x_k = \mathrm{argmin}_x \; f + x^T \nabla_x f + x^T D^2_{xy} f \; y + y^T \nabla_y f + \frac{1}{2\eta} x^T x$$

$$y_{k+1} - y_k = \mathrm{argmin}_y \; g + x^T \nabla_x g + x^T D^2_{xy} g \; y + y^T \nabla_y g + \frac{1}{2\eta} y^T y$$

- Local approximation is interactive!

- Does it have a Nash equilibrium?

Solving for the Nash Equilibrium

- Theorem:
  Among all (possibly randomized) strategies with finite first moment, the local game has a unique Nash equilibrium given by
  $$x = -\eta\left(\mathrm{Id} - \eta^2 D^2_{xy} f \, D^2_{yx} g\right)^{-1}\left(\nabla_x f - \eta D^2_{xy} f \, \nabla_y g\right)$$
  $$y = -\eta\left(\mathrm{Id} - \eta^2 D^2_{yx} g \, D_{xy} f\right)^{-1}\left(\nabla_y g - \eta D^2_{yx} g \, \nabla_x f\right)$$
- Unique equilibrium as update rule.

Competitive Gradient Descent

- Algorithm (Competitive Gradient Descent):
  At each step, compute (x_{k+1}, y_{k+1}) *from* (x_k, y_k) *as*

$$x_{k+1} - x_k = -\eta\left(\mathrm{Id} - \eta^2 D^2_{xy} f \, D^2_{yx} g\right)^{-1}\left(\nabla_x f - \eta D^2_{xy} f \, \nabla_y g\right)$$
$$y_{k+1} - y_k = -\eta\left(\mathrm{Id} - \eta^2 D^2_{yx} g \, D_{xy} f\right)^{-1}\left(\nabla_y g - \eta D^2_{yx} g \, \nabla_x f\right)$$

- Natural generalization of gradient descent to competitive optimization.

Why bilinear makes sense

- Bilinear approximation lies "between first and second order approximation"
- The right notion:
  - It is not rotationally invariant over the full space, but only rotationally invariant within the strategy space of each player
  - It preserves competitive nature of game
  - Many competitive optimization problems arise as $f(x,y) = \Phi\big(X(x), Y(y)\big) = -g(x,y)$ for a smooth and simple Φ and possibly much less regular X, Y, for example neural nets $D^2_{xy}f, D^2_{yx}f$ only involve first derivatives of X, Y
  - It works!

Caltech

What I think that they think that I think...

- We can also write the update rule as

$$\begin{pmatrix} x_{k+1} - x_k \\ y_{k+1} - y_k \end{pmatrix} = -\eta \begin{pmatrix} \mathrm{Id} & \eta D^2_{xy} f \\ \eta D^2_{yx} g & \mathrm{Id} \end{pmatrix}^{-1} \begin{pmatrix} \nabla_x f \\ \nabla_y g \end{pmatrix}$$

- For $\lambda_{\max}(A) < 1$ we can write

$$(\mathrm{Id} - A)^{-1} = \sum_{k=0}^{\infty} A^k$$

- First partial sum: Optimal if other player stays constant.
- Second partial sum: Optimal if other player thinks the other player stays constant
- Third partial sum: Optimal if other player thinks the other player thinks the other player stays constant …

Caltech

What I think that they think that I think...

- For η small enough, we recover Nash equilibrium in the limit.
- Could use Neuman series to solve matrix inverse, recovering higher order LOLA (Foerster et al., 2018)
- Krylov subspace methods more efficient!

Comparison to existing methods

- Compare to:
  - Alternating Gradient Descent (GDA)
  - Linearized Competitive Gradient Descent (LCG)
  - Symplectic Gradient Adjustment (SGA)
  - Consensus Optimization (ConOpt)
  - Optimism (OGD)

$$
\begin{array}{llllll}
\text{GDA:} & \Delta x = & -\nabla_x f & & \\
\text{LCGD:} & \Delta x = & -\nabla_x f & -\eta D^2_{xy} f \nabla_y f & \\
\text{SGA:} & \Delta x = & -\nabla_x f & -\eta D^2_{xy} f \nabla_y f & \\
\text{ConOpt:} & \Delta x = & -\nabla_x f & -\eta D^2_{xy} f \nabla_y f & -\eta D^2_{xx} f \nabla_x f \\
\text{OGDA:} & \Delta x \approx & -\nabla_x f & -\eta D^2_{xy} f \nabla_y f & +\eta D^2_{xx} f \nabla_x f \\
\text{CGD:} & \Delta x = \left(\mathrm{Id} + \eta^2 D^2_{xy} f D^2_{yx} f\right)^{-1} & \left(-\nabla_x f \right. & \left. -\eta D^2_{xy} f \nabla_y f \right. & \left. \right)
\end{array}
$$

Caltech

Comparison to existing methods

- Competitive term helps converge in bilinear game, when step size $\eta \lesssim \frac{1}{\sigma_{\max}(D^2_{xy} f)}$.
- Equilibrium term allows convergence for arbitrarily strong interactions between players.

$$
\begin{array}{lll}
\text{GDA:} & \Delta x = & -\nabla_x f \\
\text{LCGD:} & \Delta x = & -\nabla_x f \quad \boxed{-\eta D^2_{xy} f \nabla_y f} \\
\text{SGA:} & \Delta x = & -\nabla_x f \quad \boxed{-\eta D^2_{xy} f \nabla_y f} \\
\text{ConOpt:} & \Delta x = & -\nabla_x f \quad \boxed{-\eta D^2_{xy} f \nabla_y f} \quad -\eta D^2_{xx} f \nabla_x f \\
\text{OGDA:} & \Delta x \approx & -\nabla_x f \quad \boxed{-\eta D^2_{xy} f \nabla_y f} \quad +\eta D^2_{xx} f \nabla_x f \\
\text{CGD:} & \Delta x = & \boxed{(\mathrm{Id} + \eta^2 D^2_{xy} f D^2_{yx} f)^{-1}} \left(-\nabla_x f \quad \boxed{-\eta D^2_{xy} f \nabla_y f} \right)
\end{array}
$$

Caltech

Comparison to existing methods

- Consensus term prefers small-(negative sign) or large gradients (positive sign).
- Can either lead to convergence to spurious stable points (negative sign) or harm convergence (positive sign)

$$
\begin{array}{lll}
\text{GDA:} & \Delta x = & -\nabla_x f \\
\text{LCGD:} & \Delta x = & -\nabla_x f \quad -\eta D^2_{xy} f \nabla_y f \\
\text{SGA:} & \Delta x = & -\nabla_x f \quad -\eta D^2_{xy} f \nabla_y f \\
\text{ConOpt:} & \Delta x = & -\nabla_x f \quad -\eta D^2_{xy} f \nabla_y f \quad \boxed{-\eta D^2_{xx} f \nabla_x f} \\
\text{OGDA:} & \Delta x \approx & -\nabla_x f \quad -\eta D^2_{xy} f \nabla_y f \quad \boxed{+\eta D^2_{xx} f \nabla_x f} \\
\text{CGD:} & \Delta x = \left(\mathrm{Id} + \eta^2 D^2_{xy} f D^2_{yx} f\right)^{-1} & \left(-\nabla_x f \quad -\eta D^2_{xy} f \nabla_y f\right)
\end{array}
$$

Caltech

Convergence Results

- Zero sum case ($f = -g$)
- Define matrices

$$\overline{D} \coloneqq \left(\mathrm{Id} + \eta^2 D^2_{xy} f \, D^2_{yx} f\right)^{-1} \eta^2 D^2_{xy} f \, D^2_{yx} f$$

$$\widetilde{D} \coloneqq \left(\mathrm{Id} + \eta^2 D^2_{yx} f \, D^2_{xy} f\right)^{-1} \eta^2 D^2_{yx} f D^2_{xy} f$$

- Always positive definite, increasing with increasing with more interaction ($D^2_{xy} f$)

Caltech

Convergence Results

- Theorem

  If f is two times differentiable, convex-concave, with L- Lipschitz continuous mixed Hessian and $\eta||D^2_{xx}f||, \eta||D^2_{yy}f|| \leq 1$,

$$||\nabla_x f(x_{k+1}, y_{k+1})||^2 + ||\nabla_y f(x_{k+1}, y_{k+1})||^2 - ||\nabla_x f||^2 - ||\nabla_y f||^2 \leq$$
$$\nabla_x f^T \left(-\frac{\eta D^2_{xx} f}{2} - \bar{D} + 32L\eta^2||\nabla_x f||\right)\nabla_x f + \nabla_y f^T \left(\frac{\eta D^{2f}_{yy}}{2} - \tilde{D} + 32L\eta^2||\nabla_y f||\right)\nabla_y f$$

- Strong interaction *helps* convergence!

Caltech

A word on implementation

- Have to compute two matrix inverses!

$$x_{k+1} - x_k = -\eta\left(\mathrm{Id} - \eta^2 D^2_{xy} f \, D^2_{yx} g\right)^{-1}\left(\nabla_x f - \eta D^2_{xy} f \, \nabla_y g\right)$$

$$y_{k+1} - y_k = -\eta\left(\mathrm{Id} - \eta^2 D^2_{yx} g \, D_{xy} f\right)^{-1}\left(\nabla_y g - \eta D^2_{yx} g \, \nabla_x f\right)$$

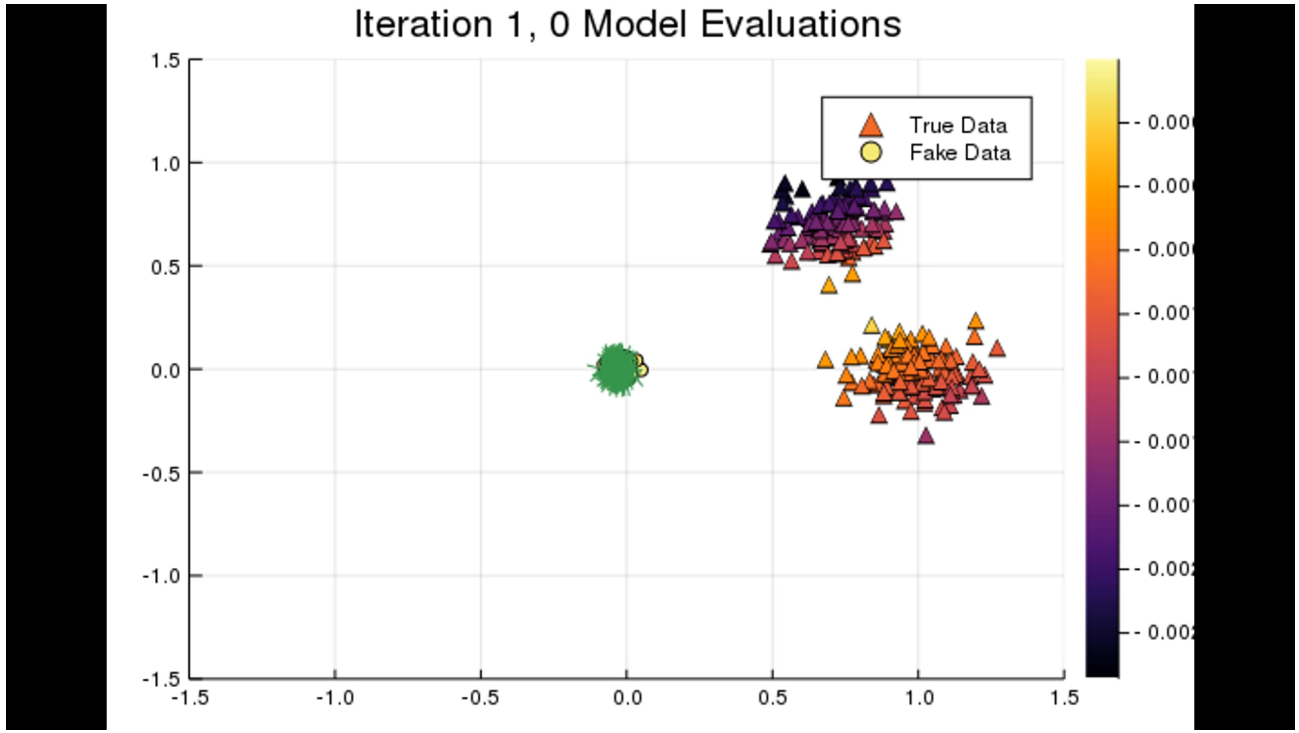
- Following Pearlmutter (1994), use mixed mode automatic differentiation to compute Hessian vector product:

$$Df(x_k, y_k)v = \frac{d}{dh}\nabla_x f(x_k, y_k + hv)\Big|_0$$

A very simple GAN

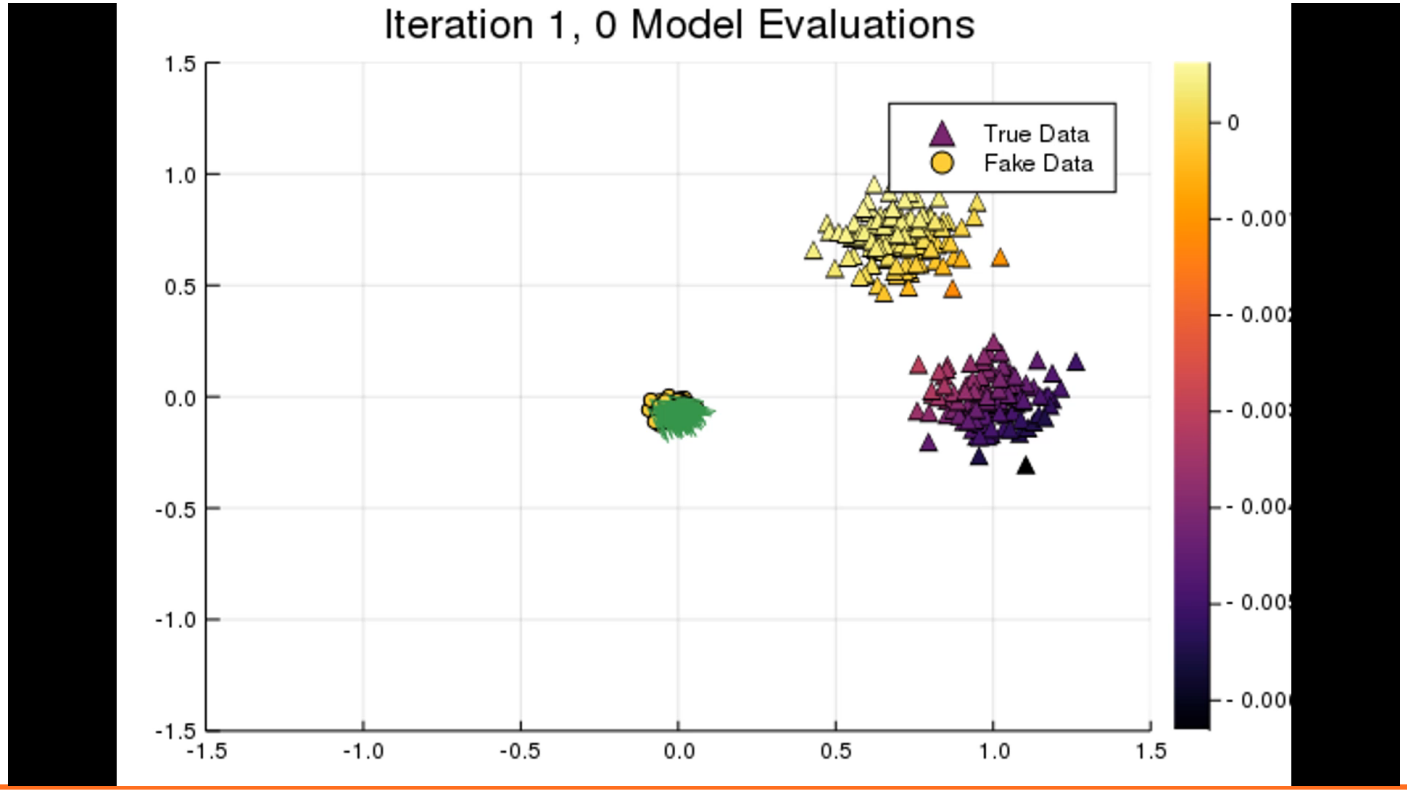
- All methods other than CGD:

(For one choice of parameters, across a range of step sizes, with RMSProp)



A very simple GAN

- CGD converges for all step sizes:



Caltech

Implicit competitive regularization

Florian Schäfer

Hongkai Zheng

A

Caltech

What is the solution of a GAN

- GAN objective for loss function L

$$\min_{\mathcal{G}} \max_{\mathcal{D}} L(\mathcal{G}, \mathcal{D})$$

- Three main interpretations in the literature

What is the solution of a GAN

- **Viewpoint 1:** *GANs defined by both players performing Gradient descent*
- Problem: Simultaneous/alternating gradient descent has poor convergence properties
- Interpretation gives little guidance as to how to improve

What is the solution of a GAN

- **Viewpoint 2:** *GANs as global min. over generator and global max. over discriminator*
- Problem: Diverges almost everywhere for finite data distributions. These games are Nashless!
- Mitigated by gradient penalties on discriminator (WGAN etc.), but requires explicit choice of distance measure.

Arora et al, Generalization and Equilibrium in GANs.
Arjovsky et al, Wasserstein GAN.
Roth et al, Stabilizing Training of GANs through Regularization

What is the solution of a GAN

- **Viewpoint 3:** *GANs find a local Nash equilibrium or some other form of local optimality*

- Problem: These points need not be good generative models and conversely, good points need not be local Nash
- No analogous results to (over-parameterized) single-agent optimization about relating local and global optima

Mazumdar et al, On Finding Local Nash Equilibria (and Only Local Nash Equilibria) in Zero-Sum Games
C. Jin et al, What is Local Optimality in Nonconvex-Nonconcave Minimax Optimization?
Berard et al, Closer Look at the Optimization Landscapes of Generative Adversarial Networks

Modeling competing agents

- **Our point of view:** *GANs via agents competing in an iterative constrained game.*
- Rather than devising new loss functions or local equilibrium concepts, embrace modeling of agent behavior and awareness

Modeling competing agents

- Need to model:
  - What information do the players have access to?
  - Which *moves* are allowed?
  - What are the player's goals?
  - How do they account for each other's actions?

An illustrative example

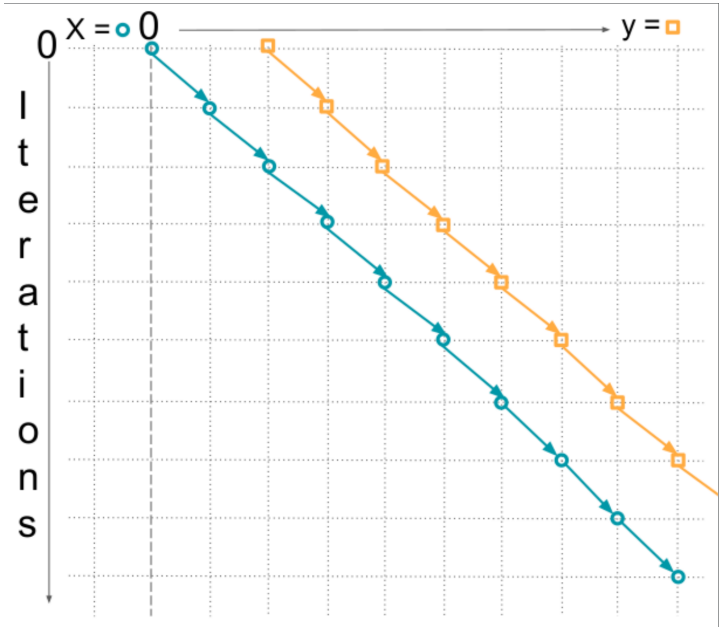
$$\min_{x} \max_{y} \; -\exp(x^2) - \alpha x y + \exp(y^2), \alpha \gg 1$$

- Does not have (local) Nash or Stackelberg equililbria.
- Does it have a meaningful solution?
- Consider three possible iterative games all of which satisfy:
- They can move by a distance of at most 1.

The global game

$$\min_{x} \max_{y} \; -\exp(x^2) - \alpha x y + \exp(y^2)$$

- Assume both players know entire loss function
- They aim to minimize their average loss (for large no. of iterations)

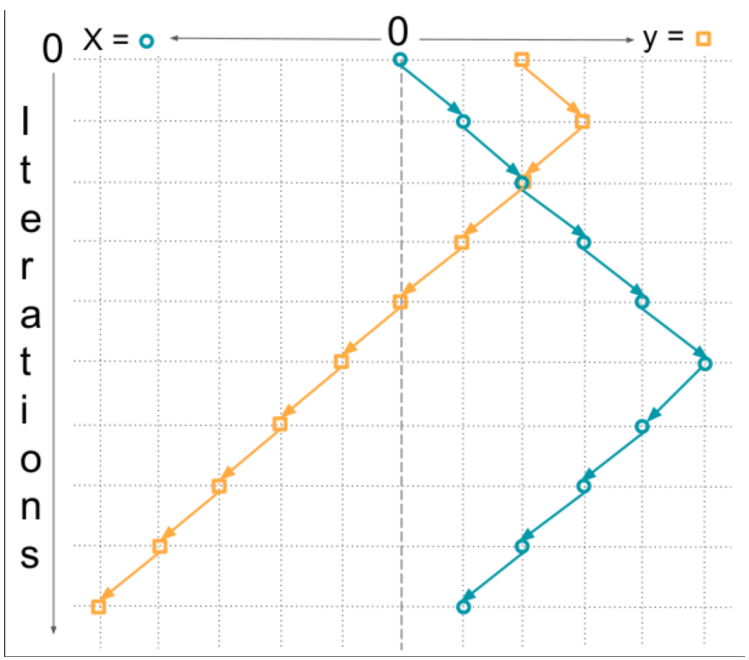


Caltech

The myopic game

$$\min_{x} \max_{y} \; -\exp(x^2) - \alpha xy + \exp(y^2)$$

- Players only know their loss within distance 1
- They aim to minimize their loss, assuming other player to stay still

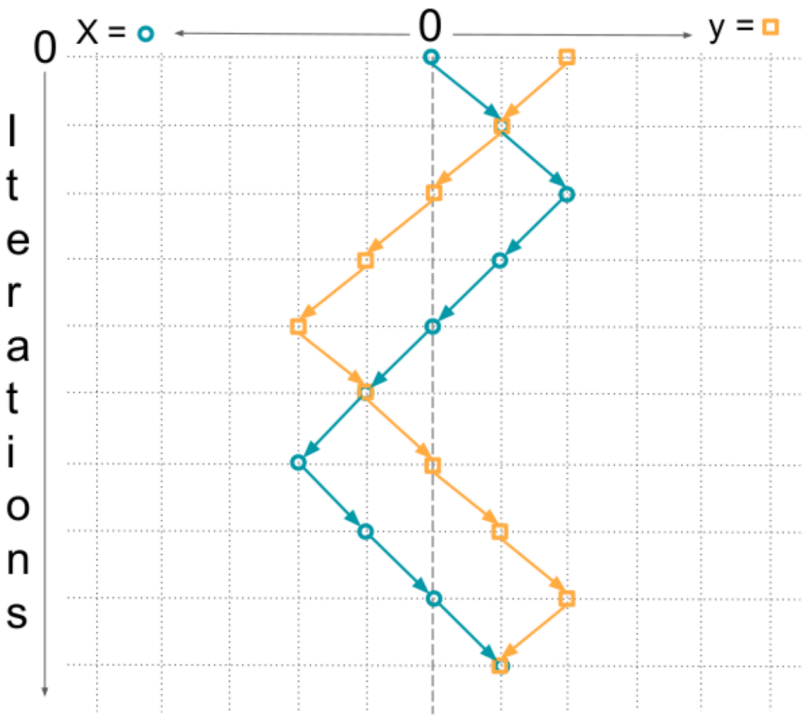


Caltech

The predictive game

$$\min_{x} \max_{y} \; -\exp(x^2) - \alpha x y + \exp(y^2)$$

- Both players know both players' loss in distance 1
- They aim to minimize their loss, aware of goals of other player
- **Implicit Competitive Regularization**



Caltech

CGD is one form of ICR

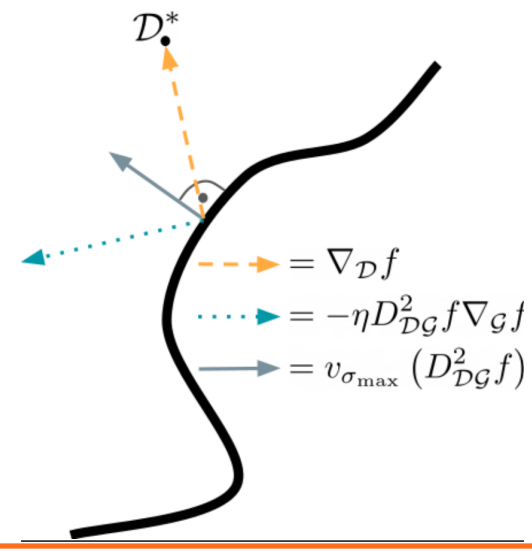
$$x_{k+1} - x_k = \mathrm{argmin}_x\ f + x^T \nabla_x f + x^T D^2_{xy} f\ y + y^T \nabla_y f + \frac{1}{2\eta} x^T x$$

$$y_{k+1} - y_k = \mathrm{argmin}_y\ g + x^T \nabla_x g + x^T D^2_{xy} g\ y + y^T \nabla_y g + \frac{1}{2\eta} y^T y$$

- Recall CGD update

$$-\eta\left(\mathrm{Id} - \eta^2 D^2_{xy} f\ D^2_{yx} g\right)^{-1}\left(\nabla_x f - \eta D^2_{xy} f\ \nabla_y g\right)$$

- Competitive term reduces other player's gradient
- Equilibrium term favors updates close to a "robust manifold"

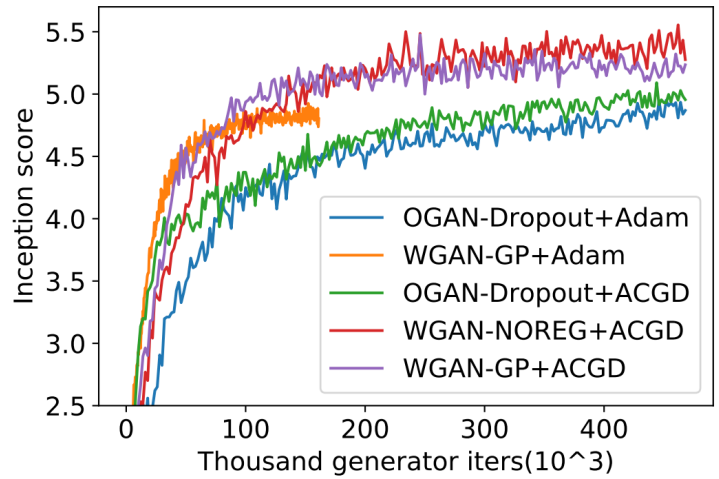


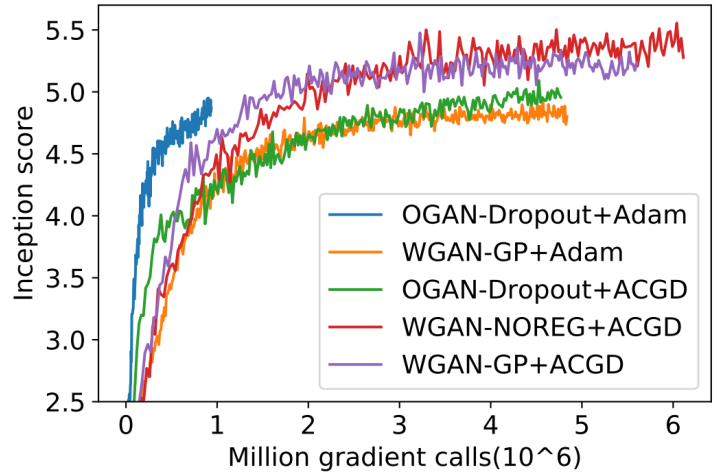
Implicit competitive regularization

- A discriminator aware of its opponent would not overtrain, to avoid retaliation of the generator
- Thus, a suitable notion of rational behavior can regularize training even for loss functions that are meaningless as minimum of a maximum
- Claim: Competitive gradient descent uses this *implicit competitive regularization* to stabilize GAN training.

Numerical results

- We use architecture intended for WGAN-GP, no additional hyperparameter tuning
- Best performance is achieved by WGAN loss with Adaptive-CGD (and no regularization)





Summary:

- New algorithm for competitive optimization
- Natural generalization of gradient descent
- Robust to strong interactions between players
- Has implicit competitive regularization: rational behavior under constraints stabilizes GAN training

- Outlook: Can we use more sophisticated modeling of agents to improve algorithms?
- Extension to multiple players: Multilinear instead of bilinear approximation

Caltech

signSGD: Compressed optimization for non-convex problems



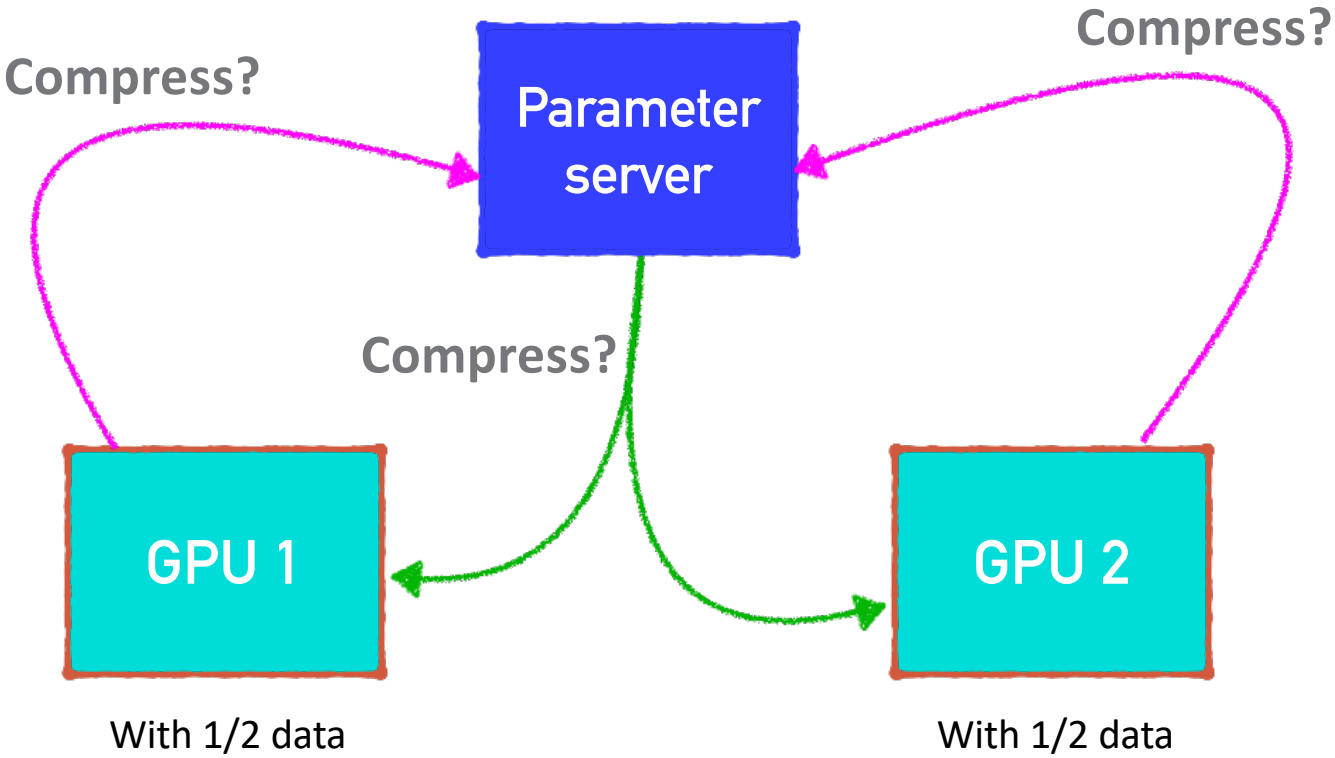


Jeremy Bernstein Jiawei Zhao Yu-Xiang Wang Kamyar Azizzade A

ICML 2018 & ICLR 2018 & ICLR 2019

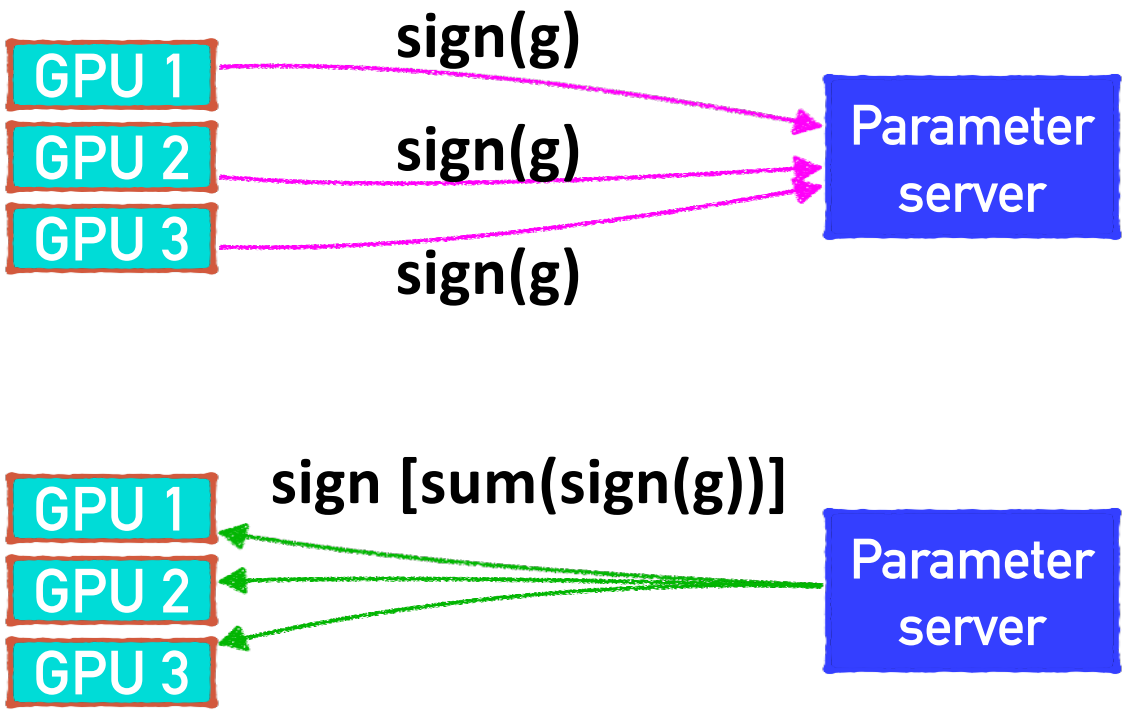
Caltech

Distributed training involves computation & communication



Caltech

Distributed training by majority vote



Caltech

Variants of SGD distort the gradient in different ways

SGD g_k

Signum $\operatorname{sign}\left(g_k + \beta g_{k-1} + \beta^2 g_{k-2} + \ldots\right)$

Adam $$\frac{g_k + \beta g_{k-1} + \beta^2 g_{k-2} + \ldots}{\sqrt{g_k^2 + \beta g_{k-1}^2 + \beta^2 g_{k-2}^2 + \ldots}}$$

Caltech

LARGE-BATCH ANALYSIS

SINGLE WORKER RESULTS

Assumptions

- Objective function lower bound f_*
- Coordinate-wise variance bound $\vec{\sigma}$
- Coordinate-wise gradient Lipschitz $\vec{L}$

Define

- Number of iterations K
- Number of backpropagations N

SGD gets rate

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K-1}\|g_k\|_2^2\right] \leq \frac{1}{\sqrt{N}}\left[2\|\vec{L}\|_\infty(f_0 - f_*) + \|\vec{\sigma}\|_2^2\right]$$

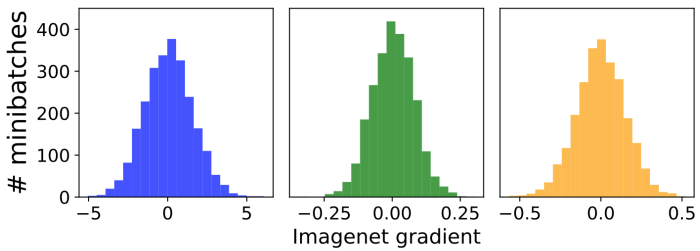
signSGD gets rate

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K-1}\sqrt{d}\|\vec{g}_k\|_2\right]^2 \leq \frac{1}{\sqrt{N}}\left[\sqrt{d}\sqrt{\|\vec{L}\|_\infty}\left(f_0 - f_* + \frac{1}{2}\right) + 2\sqrt{d}\|\vec{\sigma}\|_2\right]^2$$

Caltech

Distributed SIgnSGD: Majority vote theory

If gradients are unimodal and symmetric…



…reasonable by central limit theorem…

…majority vote with M workers converges at rate:

$$\mathbb{E}\left[\min_{0\le k\le K-1}\|g_k\|_1\right]^2$$

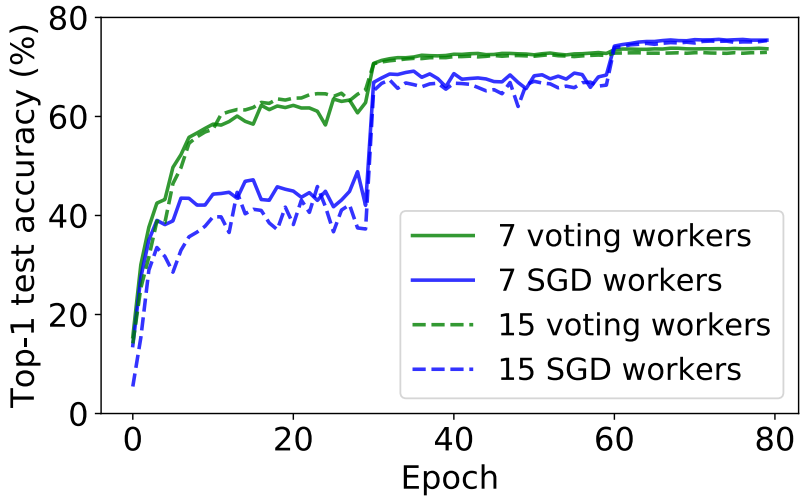
$$\le \frac{1}{\sqrt{N}}\left[\sqrt{\|\vec{L}\|_1}\left(f_0 - f_* + \frac{1}{2}\right) + \frac{2}{\sqrt{M}}\|\vec{\sigma}\|_1\right]^2$$

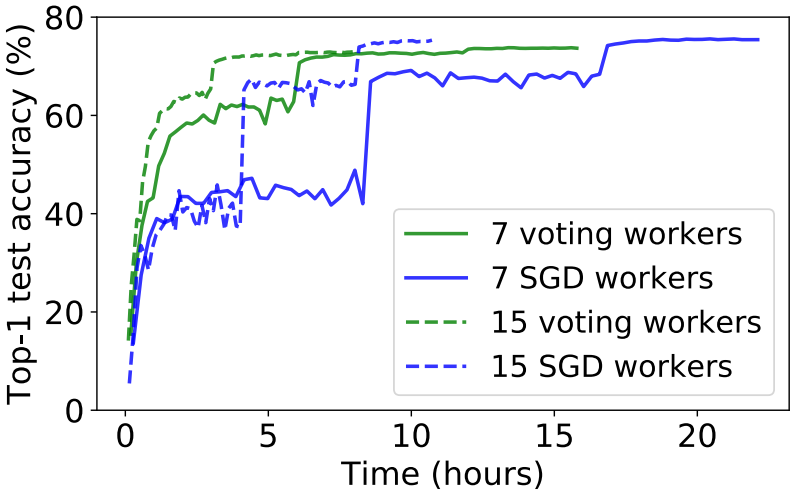
Same variance reduction as SGD

Caltech

SignSGD provides “free lunch"

P3.2x machines on AWS, Resnet50 on imagenet





Throughput gain with only tiny accuracy loss

Caltech

SignSGD on MLPerf Competition Benchmark

Object Detection Task (Mask-RCNN)

Cost of All-reduce communication
signSGD saves **81.3%** NCCL all-reduce communication cost.

Test Accuracy
signSGD only lost **3.1%** and **1.5% relative accuracy** wrt MLPerf's target performance for BBOX (Bounding Box Detection) and SEGM (Image Segmentation).

Outlook:
Integrating signSGD with apex.Amp (automatic mixed precision)

Outlook

- The last 5 years of DL: low-hanging fruits from optimization viewpoint.
- But for robust DL at scale, we need to look beyond SGD.
- A promising future for optimization theorists ;)

Caltech