Saddle Point Problems or Min-Max Problems

Pavlenko, Nikolay
n.pavlenko@innopolis.university

Dyussenova, Ashera
a.dyussenova@innopolis.university

Introduction:

Saddle point or min-max problems are more difficult problems than minimization problems. Meanwhile, min-max problems often arise in machine learning. The goal of this project is to understand the difference between min-max problems and minimization problems, and to learn methods for min-max problems.

This project is built upon an analysis of two papers: **A Variational Inequality Perspective on Generative Adversarial Networks** and **On the Convergence of Single-Cell Stochastic Extra-Gradient Methods**.

In order to comply with the requirement to have a single LaTeX file in the root directory of the project, this file will cover detailed analysis of both papers, though they will be contained in separate sections. Analysis of the problem in general and conclusion will follow the rest of the report in the final section of the .tex file.

1 A Variational Inequality Perspective on Generative Adversarial Networks:

1.1 Problem Statement:

Authors of the paper want to achieve a more stable and reliable training process for Generative adversarial networks (GANs). In pursuit of that goal they reformulate the GAN objective as a two-player game in the sense of game theory, and they address that game as a variational inequality problem

(VIP). VIP framework encompasses traditional saddle-point optimization algorithms, so it is relevant towards our investigation of min-max problems.

Once the GAN optimization problem is reformulated as VIP, the goal is to find the optimal set Ω that would verify the following equality:

$$\text{find } \omega^* \in \Omega \text{ such that } F(\omega^*)^T(\omega - \omega^*) \geq 0, \ \forall \omega \in \Omega,$$

where $\omega = (\theta, \varphi), \omega^* = (\theta^*, \varphi^*), \Omega = \Theta \times \Phi, F(\omega) = [\nabla_\theta \mathcal{L}_G(\theta, \varphi) \ \nabla_\varphi \mathcal{L}_D(\theta, \varphi)]^T$. G, D are respectively generator and discriminator in GAN, (θ, φ) are parameters of D and G, (θ^*, φ^*) is a stationary point (non-negative gradient in any direction).

1.2 Main Idea of the Approach:

Main idea of the *extrapolation from the past* technique is to optimize the existing *extrapolation* technique by storing and re-using the extrapolated gradient, thereby requiring only one computation of the gradient per update, rather than two:
Standard extrapolation:

$$\text{Compute extrapolated point: } \omega_{t+1/2} = P_\Omega[\omega_t - \eta F(\omega_{t-1/2})]$$

$$\text{Perform update step: } \omega_{t+1} = P_\Omega[\omega_t - \eta F(\omega_{t+1/2})] \text{ and store: } F(\omega_{t+1/2})$$

Extrapolation from the past:

$$\text{Compute extrapolated point: } \omega_{t+1/2} = P_\Omega[\omega_t - \eta F(\omega_t)]$$

$$\text{Perform update step: } \omega_{t+1} = P_\Omega[\omega_t - \eta F(\omega_{t+1/2})]$$

1.3 Theory and Background:

1.3.1 Related Work:

There are a multitude of VIP optimization methods studied in literature, which the authors used in order to develop their own. Two standard methods are the *gradient method*(Bruck, 1977) and the *extragradient method* (Korpelevich, 1976). More recent methods use *uniform average* in *averaging* techniques(Nedić and Ozdaglar, 2009), which guarantees convergence for any bounded monotone operator with a $O(1/\sqrt{t})$ rate, and *extragradient method* (Nesterov, 2007), which does not require averaging and converges

at an even faster rate of $O(1/t)$.

The main idea that was discovered by the authors of the paper in relation to VIP - *extrapolation from the past* was also already discovered previously by Chiang et al. (2012, Alg. 1) for general online learning.

1.3.2 Theoretical Framework:

In their work the authors use concepts from multiple different fields. Most importantly, they view the training of a GAN as a variational inequality, where the generator and discriminator aim to perform optimization in a game theory setting. They also cover stochastic optimization in their work, which was not done by many of the authors who preceded them. The solution presented in this paper is also dependent on stochastic gradient descent with averaging, extrapolation and extrapolation from the past.

1.3.3 Key Features:

- The solution presented by the authors is guaranteed to sublinearly converge in most cases, unlike some of the older methods that were also covered in the paper.

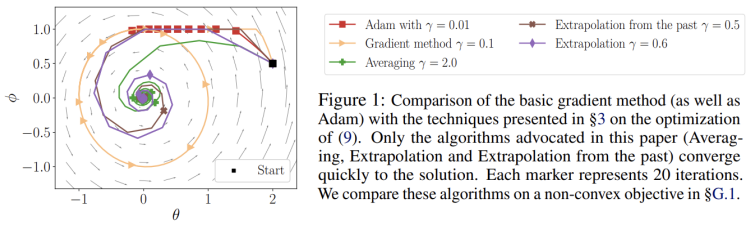


Figure 1: Comparison of the basic gradient method (as well as Adam) with the techniques presented in §3 on the optimization of (9). Only the algorithms advocated in this paper (Averaging, Extrapolation and Extrapolation from the past) converge quickly to the solution. Each marker represents 20 iterations. We compare these algorithms on a non-convex objective in §G.1.

- Recognizing that GANs involve stochastic gradients computed on mini-batches of data, the solution accounts for the inherent randomness and uncertainty in the optimization process, differing from previous work on the subject.

- The solution is not limited to a single type of GAN, and can be used with any architecture of the model of that type, making it versatile and widely applicable.

1.3.4 Causes behind Good Performance:

Several factors are worth mentioning when talking about good performance of the solution:

- Solution guarantees fewer computations per iteration when compared to original extrapolation solution, so it is by definition more computationally efficient.
- It is based upon an existing solution that was proven to work well, so as an improvement on it it is also guaranteed to perform at least as well.
- It handles stochastic gradients, which makes it exceptionally well-suited for the purpose of GAN training.

1.3.5 Improvements over Older Methods:

Authors of the paper introduce a novel technique: *extrapolation from the past*. Its improvement over the older *extrapolation* technique is in requiring only one gradient descent computation per update, rather than two.
Furthermore, their newly-developed techniques achieve a $4 - 6\%$ better inception score and the Fréchet inception distance than Miyato et al. (2018), while using CIFAR-10 dataset with a WGAN-GP and a ResNet generator.

1.4 Essence of Proof:

The essence of the proof for the method proposed in the paper lies in providing rigorous mathematical guarantees that the optimization algorithms will converge to a solution in the context of GAN training. The proof also relies on several assumptions: that function F is strongly monotone and L-Lipschitz. If those assumptions are not fulfilled, then standard extrapolation technique has to be used, as it is guaranteed to converge linearly for a bilinear game.

1.5 Experiments and Results:

Authors structured their experiments in the following way: they compared the following optimization algorithms: baselines are SGD and Adam using either simultaneous updates on the generator and on the discriminator (denoted SimAdam and SimSGD) or k updates on the discriminator alternating with 1 update on the generator (denoted AltSGDk and AltAdamk).

(simultaneous and alternating updates are different settings of the averaging technique). Variants that use extrapolation are denoted ExtraSGD and ExtraAdam (Alg. 4). Variants using extrapolation from the past are PastExtraSGD and PastExtraAdam (Alg. 4). They also presented results using as output the averaged iterates, adding Avg as a prefix of the algorithm name when we use (uniform) averaging.

1. The first test was conducted on a simple ($n = 10^3, d = 10^3$) finite sum bilinear objective constrainted to $[-1, 1]^d$, where AvgAltSGD1 and AvgPastExtraSGD performed the best:

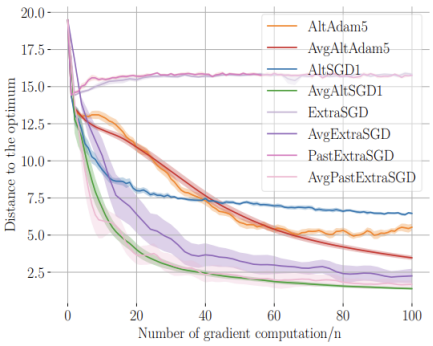


Figure 3: Performance of the considered stochastic optimization algorithms on the bilinear problem (29). Each method uses its respective optimal step-size found by grid-search.

2. The second test was conducted in the context of GAN training. Two different GAN architectures were used: WGAN and WGAN-GP. Only Adam version of gradient descent was evaluated.

   As we can see below, for both tasks, using an extrapolation step and averaging with Adam (ExtraAdam) outperformed all other methods. Combining ExtraAdam with averaging yields results that improve significantly over the previous state-of-the-art IS and FID.

Model	WGAN (DCGAN)			WGAN-GP (ResNet)		
Method	no avg	uniform avg	EMA	no avg	uniform avg	EMA
SimAdam	*6.05 ± .12*	5.85 ± .16	6.08 ± .10	*7.51 ± .17*	7.68 ± .43	7.60 ± .17
AltAdam5	*5.45 ± .08*	5.72 ± .06	5.49 ± .05	*7.57 ± .02*	8.01 ± .05	7.66 ± .03
ExtraAdam	**6.38 ± .09**	**6.38 ± .20**	**6.37 ± .08**	7.90 ± .11	**8.47 ± .10**	8.13 ± .07
PastExtraAdam	5.98 ± .15	6.07 ± .19	6.01 ± .11	7.84 ± .06	8.01 ± .09	7.99 ± .03
OptimAdam	*5.74 ± .10*	5.80 ± .08	5.78 ± .05	*7.98 ± .08*	8.18 ± .09	8.10 ± .06

Table 1: Best inception scores (averaged over 5 runs) achieved on CIFAR10 for every considered Adam variant. OptimAdam is the related *Optimistic Adam* (Daskalakis et al., 2018) algorithm. EMA denotes *exponential moving average* (with $\beta = 0.9999$, see Eq. 8). We see that the techniques of extrapolation and averaging consistently enable improvements over the baselines (in italic).

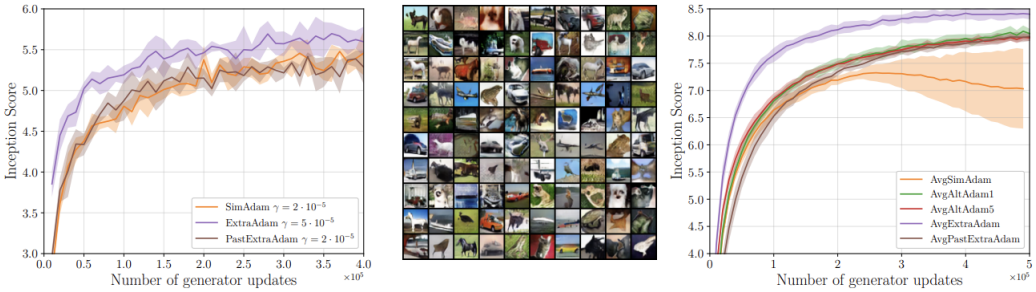


Figure 4: **Left:** Mean and standard deviation of the inception score computed over 5 runs for each method on WGAN trained on CIFAR10. To keep the graph readable we show only SimAdam but AltAdam performs similarly. **Middle:** Samples from a ResNet generator trained with the WGAN-GP objective using AvgExtraAdam. **Right:** WGAN-GP trained on CIFAR10: mean and standard deviation of the inception score computed over 5 runs for each method using the best performing learning rates; all experiments were run on a NVIDIA Quadro GP100 GPU. We see that ExtraAdam converges faster than the Adam baselines.

2 On the Convergence of Single-Cell Stochastic Extra-Gradient Methods:

2.1 Problem Statement:

Landscape of the loss function in GANs does not resemble the minimization problem, but that of a min-max game which in our words could be called a **variational inequality**. They have gained significant attention in machine learning, offering a flexible framework beyond ordinary loss function minimization, particularly in generative adversarial networks (GANs) and deep learning systems. The optimal $O(1/t)$ convergence rate for solving smooth

monotone variational inequalities is achieved by the Extra-Gradient (EG) algorithm and its variants. These methods require two projections and two oracle calls per iteration, making them costly compared to standard methods like Forward-Backward. Reducing this cost is the main goal of the current paper.

2.2 Main Idea of the Approach:

The primary focus of this research is to investigate and quantify the convergence behavior of single-call stochastic extra-gradient methods when solving variational inequalities. By understanding the underlying mechanisms of these algorithms, the paper aims to provide valuable insights into their applicability and efficiency, especially in the realm of non-monotone variational inequalities.

2.3 Theory and Background:

Variational inequalities, encompassing diverse problems like minimization, saddle-points, Nash equilibria, and fixed-point problems, serve as a foundational framework in optimization. The EG algorithm, with its variants, has been pivotal in solving monotone variational inequalities. However, the computational cost involved has led to the exploration of single-call variants, sparking a significant area of research. This paper delves into these single-call methods, studying their properties and convergence rates in both deterministic and stochastic scenarios.

2.3.1 Related Work:

Existing research has explored various algorithms, such as Forward-Backward-Forward (FBF) and gradient extrapolation mechanisms like Popov's modified Arrow-Hurwicz algorithm, to reduce the number of oracle calls and projections. Among these, single-call extra-gradient methods, including Past Extra-Gradient (PEG), Reflected Gradient (RG), and Optimistic Gradient (OG), have gained prominence for their efficiency in approximating missing gradients while making a single oracle call per iteration.

2.3.2 Theoretical Framework:

In the context of variational inequalities, the Extra-Gradient algorithm involves two projections and oracle calls per iteration. Single-call variants like PEG, RG, and OG perform a single oracle call while approximating missing

gradients differently. These algorithms converge at the optimal O(1/t) rate under certain assumptions.

2.3.3 Key Features:

- Variational inequalities provide a flexible framework in machine learning beyond traditional loss function minimization.
- Extra-Gradient and its variants achieve optimal O(1/t) convergence rates.
- Single-call extra-gradient methods, such as PEG, RG, and OG, approximate missing gradients with a single oracle call per iteration.

2.3.4 Causes behind Good Performance:

The performance of single-call extra-gradient methods is attributed to their ability to anticipate the landscape of the problem by approximating missing gradients while making only one oracle call per iteration. These methods achieve optimal convergence rates in both deterministic and stochastic variational inequalities under suitable conditions.

2.3.5 Improvements over Older Methods:

Single-call extra-gradient methods significantly reduce the computational cost associated with multiple oracle calls and projections. They retain the anticipatory properties of the original Extra-Gradient algorithm while making efficient use of oracle resources.

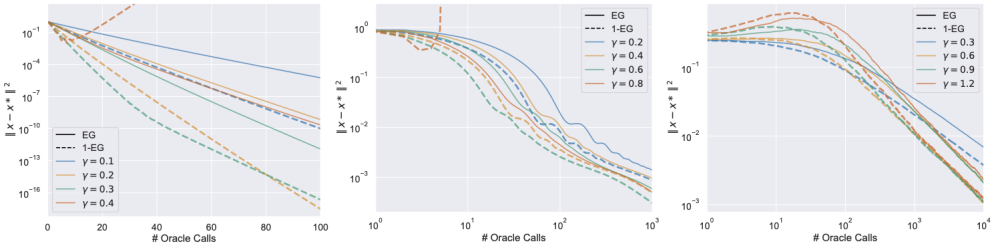
2.4 Essence of Proof:

The proofs for the convergence rates of single-call extra-gradient methods involve intricate analyses. Techniques such as quasi-descent inequalities and probabilistic arguments are employed to establish convergence properties. For stochastic variational inequalities, careful consideration of noise realizations and conditional bias is necessary to prove convergence rates.

2.5 Experiments and Results:

Experimental results demonstrate the efficiency of single-call extra-gradient methods in both deterministic and stochastic variational inequalities. These methods consistently outperform the traditional Extra-Gradient algorithm

in terms of convergence speed, especially in scenarios where computational resources are limited.



(a) Strongly monotone [$\epsilon_1 = 1, \epsilon_2 = 0$], deterministic, last iterate

(b) Monotone [$\epsilon_1 = 0, \epsilon_2 = 1$], deterministic, ergodic averaging

(c) Non-monotone [$\epsilon_1 = 1, \epsilon_2 = -1$], iid $Z \sim \mathcal{N}(0, .01)$, last iterate ($b = 15$)

Figure 1: Illustration of the performance of EG and 1-EG in the (a priori non-monotone) saddle-point problem $\mathcal{L}(\theta, \phi) = 2\epsilon_1\theta^T A_1\theta + \epsilon_2(\theta^T A_2\theta)^2 - 2\theta_1\phi^T B_1\phi - \epsilon_2(\phi^T B_2\phi)^2 + 4\theta^T C\phi$

3 Conclusion:

Both techniques that we have covered have used theoretical transformation of the problem to variational inequality problem, and then focused on optimizing the objective in the new terms. Both of them have managed to achieve better computational efficiency than older algorithms, and achieved best results during the evaluation. Both of the papers focused on this problem in order to improve GAN training. However, the experimentation of the authors of the first paper was more rigorous - more expansive tests were conducted using different settings to prove that their method works well.