A Closer Look at the Optimization Landscapes of Generative Adversarial Networks

Hugo Berard[*]
Mila, Université de Montréal
Facebook AI Research

Gauthier Gidel[*]
Mila, Université de Montréal
Element AI

Amjad Almahairi
Element AI

Pascal Vincent[†]
Mila, Université de Montréal
Facebook AI Research

Simon Lacoste-Julien[†]
Mila, Université de Montréal
Element AI

Abstract

In this work, we try to gain insights into the optimization of GANs by analyzing the game vector field resulting from concatenating the gradient of both players. We observe that the training of GANs suffers from a rotational behavior around the locally stable stationary points which can hurt convergence, and that GAN training converges to stable stationary points which is not a local Nash equilibria.[1]

1 Introduction

Deep neural networks have exhibited a large success in many applications [Krizhevsky et al., 2012]. This success has motivated many studies of their non-convex loss landscape [Choromanska et al., 2015, Kawaguchi, 2016, Li et al., 2018], which, in turn, has led to many improvements, such as introducing better initialization and optimization methods [Glorot and Bengio, 2010, Kingma and Ba, 2015]. However, only little is known about the optimization landscape of two-player games, where two players compete against each other by minimizing a different, non-convex objective. In this paper, we focus on the optimization landscape of a particular type of non-convex two-player games, namely, generative adversarial networks (GANs) [Goodfellow et al., 2014]. We attempt to clarify to what extent their optimization landscapes are intrinsically different from the standard loss surfaces that are common, for instance, in supervised learning tasks. The core questions we want to address can be summarized as:

Is the landscape of GANs different from standard loss surfaces of deep networks, and do existing training methods find local Nash equilibria?

We experimentally show that the landscape of GANs is fundamentally different from the standard loss surfaces of deep networks. Furthermore, we provide evidence that existing GAN training methods do not converge to a local Nash equilibrium.

[*]Equal contributions. Correspondence to `firstname.lastname@umontreal.ca`.
[†]Canada CIFAR AI Chair
[1]Code availbale at `https://bit.ly/2kwTu87`

Abstract for: Smooth Games Optimization and Machine Learning Workshop (NeurIPS 2019), Vancouver, Canada.

2 A Vector Field formulation of GANs

In practice, GANs are trained using first order methods that compute gradients of the losses of both players, i.e., the generator and discriminator. Following Gidel et al. [2019], we consider the players' parameters $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ as a joint state $\boldsymbol{\omega} := (\boldsymbol{\theta}, \boldsymbol{\varphi})$, and study the vector field associated with their gradients,[2] which we call the *game vector field*

$$\boldsymbol{v}(\boldsymbol{\omega}) := \begin{bmatrix} \nabla_{\boldsymbol{\theta}} \mathcal{L}_G(\boldsymbol{\omega})^\top & \nabla_{\boldsymbol{\varphi}} \mathcal{L}_D(\boldsymbol{\omega})^\top \end{bmatrix}^\top . \tag{1}$$

Verhulst [1989, Theorem 7.1] defines a LSSP $\boldsymbol{\omega}^*$ using the eigenvalues of the Jacobian of the game vector field $\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)$ at that point.

Definition 1 (LSSP)**.** *A point $\boldsymbol{\omega}^*$ is a* locally stable stationary point *(LSSP) iff*

$$\boldsymbol{v}(\boldsymbol{\omega}^*) = 0 \quad \textit{and} \quad \Re(\lambda) > 0\,, \ \forall \lambda \in \mathrm{Sp}(\nabla \boldsymbol{v}(\boldsymbol{\omega}^*))\,. \tag{2}$$

where $\Re$ denote the real part of the eigenvalue λ belonging to the spectrum of $\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)$.

This definition is not easy to interpret but one can intuitively understand a LSSP as a stationary point (a point $\boldsymbol{\omega}^*$ where $\boldsymbol{v}(\boldsymbol{\omega}^*) = 0$) to which all neighbouring points are attracted.

Another type of stationary points often considered in the literature is the LNE.

Definition 2 (LNE)**.** *A point $\boldsymbol{\omega}^*$ is a* Local Nash equilibrium *(LNE) iff*

$$\|\boldsymbol{v}(\boldsymbol{\omega}^*)\| = 0\,, \ \nabla^2_{\boldsymbol{\theta}} \mathcal{L}_G(\boldsymbol{\omega}^*) \succ 0 \textit{ and } \nabla^2_{\boldsymbol{\varphi}} \mathcal{L}_D(\boldsymbol{\omega}^*) \succ 0$$

where $\boldsymbol{S} \succ 0$ if and only if $\boldsymbol{S}$ is definite positive.

LNE can be understood as a point such that no players can improve it's own loss function. When $\mathcal{L}_G(\boldsymbol{\omega}) = -\mathcal{L}_D(\boldsymbol{\omega})$, being a LNE is a sufficient condition for a point to be a LSSP [Mazumdar and Ratliff, 2018]. However, some LSSPs may not be LNEs [Adolphs et al., 2018]. In the following section we propose tools to study the game vector field around stationary points it has converged to, and analyze whether we have converged to a LNE or a LSSP.

3 Proposed visualization: Path-angle

To study the landscape of the game vector field around stationary points, we propose to look at some metrics along the linear path between parameters early in learning and parameters late in learning. In particular we propose the following metrics:

Path-angle. We first ensure that we are in a neighborhood of a stationary point by computing the norm of the vector field (1).

Once we are close to a final point $\boldsymbol{\omega}'$, i.e., in a neighborhood of a LSSP, we propose to look at the angle between the vector field and the linear path from $\boldsymbol{\omega}$ to $\boldsymbol{\omega}'$. Specifically, we monitor the cosine of this angle, a quantity we call *Path-angle*:

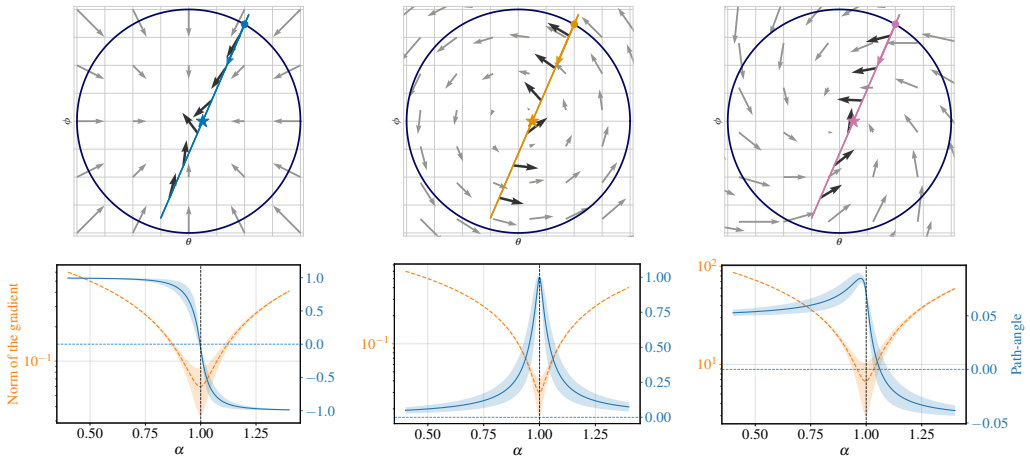
$$c(\alpha) := \frac{\langle \boldsymbol{\omega}' - \boldsymbol{\omega}, \boldsymbol{v}_\alpha \rangle}{\|\boldsymbol{\omega}' - \boldsymbol{\omega}\| \|\boldsymbol{v}_\alpha\|}\,, \quad \boldsymbol{v}_\alpha := \boldsymbol{v}(\alpha \boldsymbol{\omega}' + (1-\alpha)\boldsymbol{\omega})\,, \tag{3}$$

where $\alpha \in [a, b]$. Usually $[a, b] = [0, 1]$, but since we are interested in the landscape around a LSSP, it might be more informative to also consider further extrapolated points around $\boldsymbol{\omega}'$ with $b > 1$.

Eigenvalues of the Jacobian. Another important tool to gain insights on the behavior close to a LSSP, as discussed in §2, is to look at the eigenvalues of $\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)$. We propose to compute the top-20 eigenvalues of this Jacobian. When all the eigenvalues have positive real parts, we conclude that we have reached a LSSP, and if some eigenvalues have large imaginary parts, then the game has a strong rotational behavior. Similarly, we can also compute the top-20 eigenvalues of the diagonal blocks of the Jacobian, which correspond to the Hessian of each player. These eigenvalues can inform us on the type of stationary points we have reached, in particular do we reach a LNE or a LSSP.

[2] Note that, in principle, the joint vector field (1) is *not* a gradient vector field, i.e., it cannot be rewritten as the gradient of a single function.

We can distinguish three different archetypal behavior of the game vector field around a LSSP. We show the behavior of Path-angle for each of them in Figure 1(a-c): *attraction only* when the vector field perfectly points to the optimum (Fig. 1a); *rotation only* when the vector field is orthogonal to the direction to the optimum and rotates around the LSSP (Fig. 1b); *General setting* when the vector field has both attraction and rotation (Fig. 1c). Figure 1 shows that Path-angle can capture and characterize the different behaviors around a LSSP.



(a) Attraction only (b) Rotation only (c) General setting

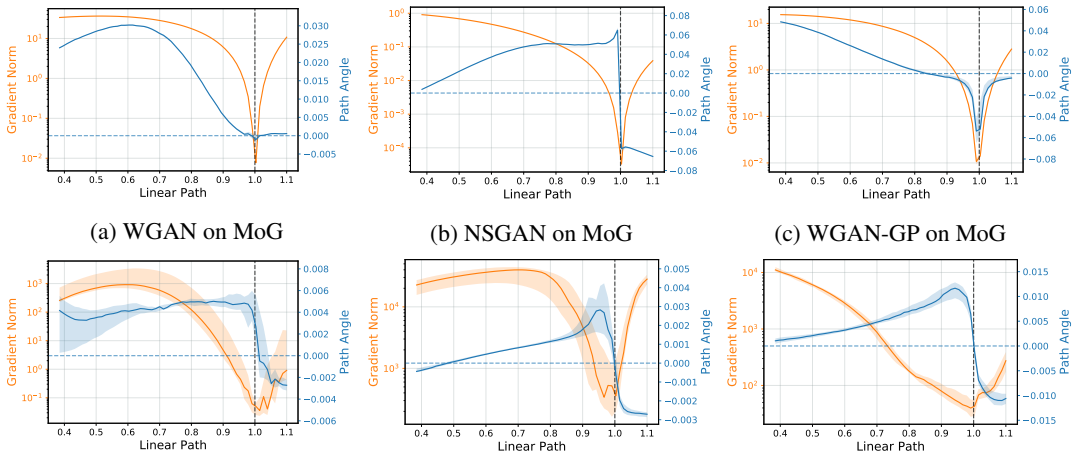
Figure 1: **Above**: game vector field for different archetypal behaviors. Grey arrows represent the vector field. The equilibrium of the game is at $(0, 0)$. Black arrows correspond to the directions of the vector field at different linear interpolations between two points: $\bullet$ and $\star$. **Below**: path-angle $c(\alpha)$ for different archetypal behaviors (right y-axis, in blue). The left y-axis in orange correspond to the norm of the gradients. Notice the "bump" in path-angle (close to $\alpha = 1$), characteristic of rotational dynamics.

4 Numerical results on GANs

Using the tools described in the previous section, we propose to study the landscape of a variety of GAN models trained on two datasets. We first propose to train a GAN on a toy task composed of a 1D mixture of 2 Gaussians (MoG) with 10,000 samples. For this task both the generator and discriminator are neural networks with 1 hidden layer and ReLU activations. We also train a GAN on MNIST, where we use the DCGAN architecture [Radford et al., 2016] where we replace batch normalization [Ioffe and Szegedy, 2015] by spectral normalization [Miyato et al., 2018] for improved stability of training. We focus on two common GAN loss formulations: we consider both the original non-saturating GAN (NSGAN) formulation proposed in Goodfellow et al. [2014] and the WGAN-GP objective described in Gulrajani et al. [2017].

Evidence of rotation around locally stable stationary points in GANs We first look at the path-angle and at the eigenvalues of the Jacobian of the game vector field (Fig. 2). We observe a combination of a bump and a sign switch similar to Fig. 1c. Also Fig. 3c clearly shows the existence of imaginary eigenvalues with large magnitude. Those observations clearly show the existence of rotational behaviour around the LSSP. This is an important observation since it was shown that rotational behavior around LSSP can lead to instabilities [Gidel et al., 2019].

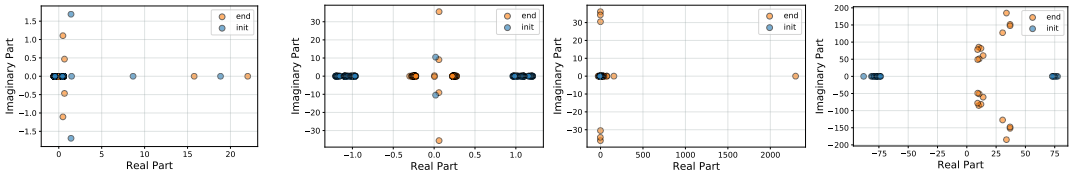
The locally stable stationary points of GANs are not local Nash equilibria Fig. 3c shows that the top 20 eigenvalues all have positive real parts, this indicates that those points are locally stable. However when looking at the eigenvalues of the hessian of both players we observe that the generator is not at a minimum but instead is only at a saddle point. Thus we have converged to a LSSP which is not a local Nash equilibrium. This observation was consistent across experiments. It is an open question wheter it is actually better to converge to a LNE than a LSSP, we leave this as future work.



(a) WGAN on MoG (b) NSGAN on MoG (c) WGAN-GP on MoG

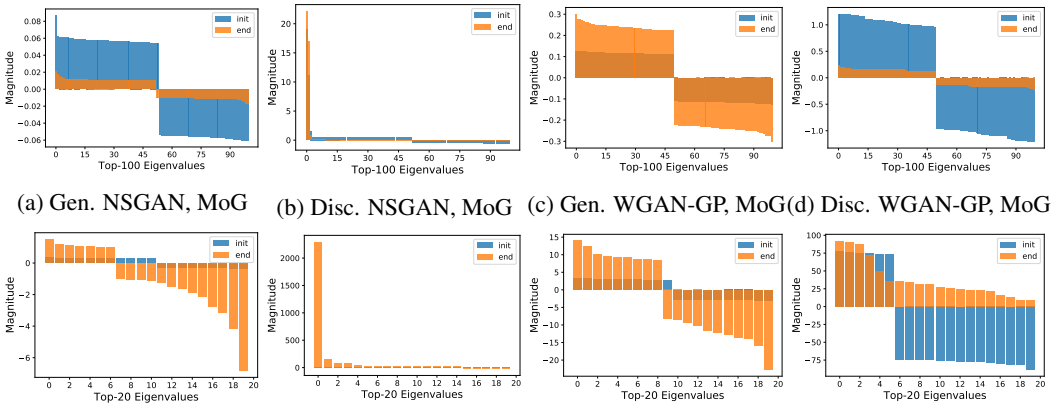
(d) Disc. minimization, MNIST (e) NSGAN on MNIST, IS: 8.97 (f) WGAN-GP on MNIST, IS: 9.46

Figure 2: Path-angle computed on MoG and MNIST. For MoG the ending point is a generator which has learned the distribution. For MNIST we indicate the Inception score (IS) at the ending point of the interpolation. Notice the "bump" in path-angle (close to $\alpha = 1.0$), characteristic of games rotational dynamics, and absent in the minimization problem (d).



(a) NSGAN on MoG (b) WGAN-GP on MoG (c) NSGAN on MNIST (d) WGAN-GP on MNIST

Figure 3: Eigenvalues of the Jacobian of the game computed on MoG and MNIST. Large imaginary eigenvalues are characteristic of rotational behavior. Notice that NSGAN and WGAN-GP objectives lead to very different landscapes (see relative scale of imaginary v.s. real part). The much larger imaginaries in WGAN-GP could be responsible for it taking much longer to reach a LSSP.



(a) Gen. NSGAN, MoG (b) Disc. NSGAN, MoG (c) Gen. WGAN-GP, MoG (d) Disc. WGAN-GP, MoG

(e) Gen. NSGAN, MNIST (f) Disc. NSGAN, MNIST (g) G. WGAN-GP, MNIST (h) D. WGAN-GP, MNIST

Figure 4: Top k-Eigenvalues of the Hessian of each player (in terms of magnitude) in descending order for NSGAN and WGAN-GP trained on MoG and MNIST. Eigenvalues indicate that the Generator does not reach a local minimum but a saddle point. Thus the training algorithms do not converge to a Nash equilibrium.

Acknowledgments.

This research was partially supported by the Canada CIFAR AI Chair Program, the Canada Excellence Research Chair in "Data Science for Realtime Decision-making", by the NSERC Discovery Grant RGPIN-2017-06936, by a Borealis AI fellowship and by a Google Focused Research award. The authors would like to thanks Tatjana Chavdarova for fruitful discussions.

References

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

A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun. The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, 2015.

G. Gidel, H. Berard, P. Vincent, and S. Lacoste-Julien. A variational inequality perspective on generative adversarial nets. *ICLR*, 2019.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein GANs. In *NeurIPS*, 2017.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

K. Kawaguchi. Deep learning without poor local minima. In *NeurIPS*, 2016.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

J. Li, A. Madry, J. Peebles, and L. Schmidt. On the limitations of first order approximation in gan dynamics. In *ICML*, 2018.

E. Mazumdar and L. J. Ratliff. On the convergence of gradient-based learning in continuous games. *ArXiv*, 2018.

T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

F. Verhulst. *Nonlinear differential equations and dynamical systems*. Springer Science & Business Media, 1989.