Journal of Machine Learning Research 1 (2000) 1-48 Submitted 4/00; Published 10/00

The Dynamics of GANs

Ruoyi Wang RW2676@NYU.EDU
Courant Institute of Mathematical Sciences
New York University
251 Mercer Street, New York, NY 10012

Zijian Liu ZL3067@NYU.EDU
Courant Institute of Mathematical Sciences
New York University
251 Mercer Street, New York, NY 10012

Editor:

Abstract

In this survey, we review recent theoretical advances on the dynamics of GANs. We organize our review loosely by their approaches and assumptions, logically ordered by how some work gives rise to others, to highlight their motivation, deficiencies, and connections. We intentionally covered a wide variety of approaches to show the diverse point of view people are taking in this area, and hope to give readers a complete picture of this area in a compact way.

Keywords: GANs, min-max optimization, dynamics, saddle points

1. Introduction

Since the debut of generative adversarial networks (GANs) (Goodfellow et al. (2014)), it has become one of the most popular and successful generative models. However, GANs are notorious for their training difficulty, and their training dynamics are poorly understood. For instance, GANs are typically analyzed from the *divergence minimization* perspective (Nowozin et al. (2016); Arjovsky et al. (2017)), but their convergence guarantees are rarely delivered in practice, and (Fedus et al. (2018)) pointed out many discrepancies between these theories and practice.

Therefore, it is worthwhile to directly study the dynamics of GANs. Works along this line focus on three main problems: existence of a stable equilibrium, improved convergence to a equilibrium, and selective convergence to a good equilibrium. To deal with these problems, three main approaches are commonly adopted: regularization, gradient flow, and mean-field analysis.

2. Notations

In this survey, $\|\cdot\|$ means Euclidean norm for vector. For matrix, it means spectral norm. For a matrix W, $\sigma_1(W)$ means the largest singular value of it. $\|f\|_L$ means the Lipschitz constant of f. $\mathcal{C}^n$ means the class which contains functions with n continuous derivatives. $\delta_\theta(x) = \delta(x-\theta)$, where $\delta(x)$ is Dirac delta function. In a GANs, we use G and D to represent the generator and the discriminator, θ_g and θ_d to represent the parameters of them. θ means (θ_g, θ_d). $\mathbb{P}$ means a probability distribution, if such probability distribution is continuous, denote its probability density function as p.

3. Aren't the problems already solved?

In the original paper of GANs (Goodfellow et al. (2014)), the training of GANs are described as a process of divergence minimization, where it has been proven that when the training converges, the distribution of the generated data, $\mathbb{P}_{gen}$, would converge to the latent distribution $\mathbb{P}_{data}$ that have generated the training data, under the assumptions that the discriminator can be trained to optimal after each generator update. However, a closer investigation will reveal several major problems in this proof:

1. This proof does not say whether the training will converge at all. Indeed, it has been pointed out repeatedly that a min-max game between two players may not always converge. A simple example would be rock-scissor-paper between two players.
2. This proof assumes the discriminator is trained to the optimal after each generator update, which does not reflect our practice in training GANs.
3. This proof assumes we can sample from the true data distribution $\mathbb{P}_{data}$, where in practice we are given only a fixed number of samples from $\mathbb{P}_{data}$.

It then can be seen that most framework based on divergence minimization, including the well-known Wasserstein GANs (Arjovsky et al. (2017)), would all suffer from the same problems. Indeed, as verified by (Fedus et al., 2018), certain results in their experiments defy the theory of divergence minimization.

This survey, then, focuses on these problems. To deal with the first the problem, we want to investigate whether stable equilibrium exists, which are discussed in Section 4. To deal with the second problem, we want a deeper understanding of the dynamics of the training process, and approaches to improve or guarantee the training process will converge to a equilibrium. These works are discussed in Section 5. To deal with the third problem, we should avoid making the assumptions that we can sample from the true data distribution, which is already implicitly incorporated in most of the analyses in Section 5. A particularly interesting work analyzed the generalization properties of these equilibrium points, explicitly taking into consideration that we have only limited data samples. This work is discussed in Section 6.

4. Stability

(Nagarajan and Kolter, 2018) deals with the immediate problem of whether a stable equilibrium exists at all in a GANs game. The problem arises because GANs optimization does not actually correspond to a convex-concave game, even for linear models.

To see that, suppose $D(x) = \theta_d x + \theta'_d, G(x) = \theta_g x + \theta'_g$. Now the GANs objective is:

$$V(G, D) = \mathop{\mathbb{E}}_{x \sim p_{data}}[f(\theta_d x + \theta'_d)] + \mathop{\mathbb{E}}_{z \sim p_{latent}}[f(-\theta_d(\theta_g z + \theta'_g) - \theta'_d)]$$

where f is a concave function depending on our choice. Therefore, it can be seen that V is concave in θ_d and θ'_d. But V is also concave in θ_g and θ'_g. In this example, GANs optimization actually corresponds to a concave-concave optimization, where most problems are not stable. It has been proven in the paper that this example is not a particular case, but holds true for virtually any real parameterization of GANs.

We will grow even more suspicious by considering the case where the generator has been trained to optimality such that $p_{gen} = p_{data}$, the optimal discriminator would output 0 on all examples. However, given the optimal discriminator, any generator will be considered optimal to the discriminator. How could this system resist a small disturbance at the equilibrium point?

We can now see the importance of the work by (Nagarajan and Kolter, 2018), where it has been proven that under certain assumptions, GANs **is** actually locally stable around certain equilibrium points. One of those key assumptions is that the magnitude of the update on the equilibrium discriminator, i.e. $\|\nabla_{\theta_d} V(\theta_d, \theta_g)|_{\theta_d = \theta^*_d}\|^2$ where θ^*_d is the value of θ_d at a equilibrium point, is strongly convex. This assumption, in conjunction with the assumption that the objective is strongly concave in the discriminator parameter space at equilibrium (it is already concave), breaks the concave-concave problem in GANs optimization, and eventually leads to the highly desirable conclusion that stable equilibrium do exist in GANs optimization, proven with the help of the linearization theorem.

This work ensured that we do have a stable equilibrium point to converge to, a result some of the work in Section 5 indirectly relies on. An interesting side note is this work proposed a regularization term $\|\nabla_{\theta_d} V(\theta_d, \theta_g)\|^2$, where we can see is reminiscent to the well-known gradient penalty. A key difference to notice is the gradient is taken w.r.t. the discriminator parameter θ_d rather than the input data. Also notice this is the term they discovered which we can assume to be convex.

5. Convergence

The proof that we do have a stable equilibrium point to converge to gives us some sense of security when training GANs, but more problem remains. In particular, how can we converge to a equilibrium in a faster and more stable way? Is the equilibrium point actually what we want? In this section we review recent theoreticcal works on these questions.

5.1 Improving convergence

We begin with some of the simpler works, which will also help us understand the more sophisticated works. We begin our discussion with the work by (Metz et al., 2016), where a closer investigation at our common practice reveals a more robust training objective and interesting insights. Then, we discuss (Roth et al., 2017), where the authors attempted to stabilize the training of GANs by introducing additive noise, and showed this method has a corresponding regularization term. We then introduce the work by (Heusel et al., 2018), where the authors introduced a two-timescale update rule for GANs optimization. As we shall see shortly, it inspired another important work.

5.1.1 Unrolled GANs

In Metz et al. (2016), the authors closely investigated the common practice that we update the discriminator k iterations after each update to the generator. Describe θ_d^k, the value of θ_d after the k-th update after the previous update to the generator, by the recurrent equation:

$$\begin{aligned}\theta_d^0 =& \theta_d \\ \theta_d^{k+1} =& \theta_d^k + \lambda \frac{\mathrm{d}V(\theta_g, \theta_d^k)}{\theta_d^k}\end{aligned}$$

where θ_g and θ_d are the parameters of the generator and discriminator respectively after the previous update to the generator, V is the objective for the discriminator. Now, when we update the generator, we are actually using

$$V_k(\theta_g, \theta_d) \triangleq V(\theta_g, \theta_d^k(\theta_g, \theta_d))$$

where f is the objective function of the generator. Calculating the gradient of f_K w.r.t. θ_g, we get:

$$\frac{\mathrm{d}V_k(\theta_g, \theta_d)}{\mathrm{d}\theta_g} = \frac{\partial V_k(\theta_g, \theta_d^k(\theta_g, \theta_d))}{\partial \theta_g} + \frac{\partial V_k(\theta_g, \theta_d^k(\theta_g, \theta_d))}{\partial \theta_d^k(\theta_g, \theta_d)} \frac{\mathrm{d}\theta_d^k(\theta_g, \theta_d)}{\mathrm{d}\theta_g}$$

We can now notice that we actually threw away the second term of the gradient in our common practice. Furthermore, the part we throw away captures how the discriminator would react to a change in the generator, discouraging the generator from collapsing. By including this term, we can improve the stability of training.

The work by (Nagarajan and Kolter, 2018), as discussed in Section 4 is based on 1-unrolled GANs.

5.1.2 Noise-induced regularization

The intuition to introduce additional noise in the training to stabilize training comes from (Arjovsky and Bottou (2017)), where it has been shown that dimensional misspecification is a major problem in GANs optimization. Namely, in early stages of training, it is very likely that $Supp(p_{gen}) \cap Supp(p_{data})$ have zero measure. In this case, it is very likely that the discriminator will saturate, leading to training failure. A way to deal with the problem is to introduce additive noise that has support everywhere.

However, adding high-dimensional noise introduces significant variance in the parameter estimation process, and is therefore advised against by (Arjovsky et al., 2017). The contribution of this work is the discovery of a regularization term equivalent to adding noise in the f-GANs setting, enabling us to deal with the dimensional misspecification problem by introducing noise without suffering from the high variance.

Notice once we use Gaussian noise to corrupt the data, we can use Talyor approximation and expectation to eliminate. Finally we have the following objective function:

$$V(G, D) = \mathop{\mathbb{E}}_{\boldsymbol{x} \sim p_{data}}[D(\boldsymbol{x})] - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim p_{gen}}[f^c(D(\boldsymbol{x}))] - \lambda \mathop{\mathbb{E}}_{\boldsymbol{x} \sim p_{gen}}[f^{c''}(D(\boldsymbol{x})) \|\nabla_{\boldsymbol{x}} D(\boldsymbol{x})\|^2]$$

where f is a convex function satisfying $f(1) = 0$ and f^c means the Fenchel's dual of f. We can interpret the last term as a regularity part.

5.1.3 Two-timescale update rule

In Heusel et al. (2018), the authors proposed a new way named Two Time-Scale Update Rule (TTUR) to make our min-max converge better. In tuition, TTUR is based on SGD, and we should obey the following formulation to update our θ_g and θ_d in the $k+1$ iterations of training:

$$\begin{aligned}\theta_g^{k+1} &= \theta_g^k + a^k(-\nabla_{\theta_g} V(\theta_g, \theta_d) + M_g^k) \\ \theta_d^{k+1} &= \theta_d^k + b^k(\nabla_{\theta_d} V(\theta_g, \theta_d) + M_d^k)\end{aligned}$$

where $\{a^k\}$ and $\{b^k\}$ satisfying $\sum a^k = \infty$, $\sum b^k = \infty$, $\sum(a^k)^2 < \infty$,$\sum(b^k)^2 < \infty$. M_g^k and M_d^k are two random variables under some constraints. The researchers show with these additional new parameters, our training is equivalent to to ODE for the Heavy Ball with Friction. So with additional parameters, GANs will converge to the local Nash equilibrium better.

5.1.4 A more specific on Wasserstein GANs

Arjovsky et al. (2017) proposed Wasserstein GANs using Wasserstein distance which has a better property than other GANs constructed by divergence ,especially when the data distribution and generator distribution without overlap. We know the objective function of Wasserstein GANs is:

$$V(D, G)_{\|D\|_L \leq 1} = \mathop{\mathbb{E}}_{\boldsymbol{x} \sim p_{data}}[D(\boldsymbol{x})] - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim p_{gen}}[D(\boldsymbol{x})]$$

However, in practice how to converge to equilibrium with a discriminator satisfying the Lipschitz condition is always a hard thing. The famous method is gradient penalty. Miyato et al. (2018) shows a different but nice way to solve this specific problem. Consider our discriminator as the following function:

$$D(\boldsymbol{x}) = W^{L+1}(a^L(W^L(\cdots a^1(W^1\boldsymbol{x})\cdots)))$$

where W^i is matrix and a^i is the activation function. $\theta_d = \{W^i : 1 \leq i \leq L+1\}$. Notice the following property of Lipschitz constant:

$$\|f \circ g\|_L \leq \|f\|_L \|g\|_L$$

For a matrix W, $\|W\|_L = \|W\| = \sigma_1(W)$. For ReLU function we have $\|\mathrm{ReLU}\|_L = 1$. If we choose ReLU function as the activation function. We can derive the following result:

$$\|D\|_L \leq (\prod_{i=1}^{L+1} \|W^i\|_L)(\prod_{i=1}^{L} \|a^i\|_L) = \prod_{i=1}^{L+1} \sigma_1(W^i)$$

Then we have a natural idea is when we update θ_d, we can use $\widetilde{W^i} = W^i/\sigma_1(W^i)$ to the right side of inequality to be 1. We call this kind of regularity is spectral normalization. We don't need more theoretical analysis of this method. Because we fit the Lipschitz constraints naturally.

5.2 Analysis with smooth games

In this section we focus on a family of works that adopted the idea of gradient flow to analyze the training behavior of GANs. Recall that the update rules of GANs are:

$$\begin{aligned}\theta_g^{k+1} =& \theta_g^k - \gamma \nabla_{\theta_g} V(\theta_g^k, \theta_d^k) \\ \theta_d^{k+1} =& \theta_d^k + \gamma \nabla_{\theta_d} V(\theta_g^k, \theta_d^k)\end{aligned}$$

Let $\gamma \to 0$, we obtain the gradient flow:

$$\dot{\theta} = -v(\theta)$$

where v is defined as the following vector field:

$$v(\theta_g, \theta_d) = \begin{bmatrix} \nabla_{\theta_g} V(\theta_g, \theta_d) \\ -\nabla_{\theta_d} V(\theta_g, \theta_d) \end{bmatrix}$$

A equilibrium in this system is then a point where $v(\theta_g^*, \theta_d^*) = 0$. We can then gain insight into the convergence to an equilibrium in GANs training by analyzing the continuous game limit, as demonstrated by (Mescheder et al., 2017).

5.2.1 The numerics of GANs

(Mescheder et al., 2017) focuses on convergence properties in the two-player smooth zero-sum game. The key observation is that the convergence properties of this game is determined by the eigenvalues of the Jacobian of the vector field:

$$v'(\theta_g, \theta_d) = \begin{bmatrix} \nabla^2_{\theta_g} V(\theta_g, \theta_d) & \nabla^2_{\theta_g, \theta_d} V(\theta_g, \theta_d) \\ -\nabla^2_{\theta_d, \theta_g} V(\theta_g, \theta_d) & -\nabla^2_{\theta_d} V(\theta_g, \theta_d) \end{bmatrix}$$

The key observation here is in fixed-point iterations, if the absolute values of the eigenvalues of the Jacobian are all smaller than 1, then the fixed-point iteration converges to the fixed point.

Finding the equilibrium point can be recast into a fixed-point problem in a similar way as Newton's method. Therefore, we can draw the conclusion that the convergence of current algorithms suffers due to two factors:

- presence of eigenvalues of the Jacobian of the gradient vector field with zero real-part.
- eigenvalues with big imaginary part.

(Mescheder et al., 2017) then proceeds to propose a slightly modified vector field, which they call consensus optimization:

$$w(\theta) = v(\theta) - \nabla L(\theta)$$

where

$$\nabla L(\theta) = v'(\theta)^T v(\theta)$$

The intuition of such method is to constrain $\frac{1}{2}\|v(\theta)\|^2$ and make it to converge to equilibrium faster.

(Mescheder et al., 2018) then uses this approach to analyze a collection of the widely-used GANs with a simple example called Dirac-GAN, which consists of a generator distribution $p_{gen} = \delta_{\theta_g}$ and a linear discriminator $D(x) = \theta_d x$. The true data distribution p_{data} is given by δ_0, and found out most of them do not converge.

The trap here, however, is that not every equilibrium is a Nash equilibrium. As pointed out in Mazumdar et al. (2019), these approaches can converge to an equilibrium that is not a Nash equilibrium. Since Nash equilibrium is what we are after, (Mazumdar et al., 2019) introduced an new gradient flow to avoid non-Nash equilibriums, and a two time-scale algorithm to discretize the gradient flow.

5.2.2 Finding only Nash equilibrium

Recall the Jacobian matrix of vector field v:

$$v'(\theta_g, \theta_d) = \begin{bmatrix} \nabla^2_{\theta_g} V(\theta_g, \theta_d) & \nabla^2_{\theta_g,\theta_d} V(\theta_g, \theta_d) \\ -\nabla^2_{\theta_d,\theta_g} V(\theta_g, \theta_d) & -\nabla^2_{\theta_d} V(\theta_g, \theta_d) \end{bmatrix}$$

where we always have $\nabla^2_{\theta_d,\theta_g} V(\theta_g, \theta_d) = (\nabla^2_{\theta_g,\theta_d} V(\theta_g, \theta_d))^T$.

Also recall that a equilibrium point (θ_g^*, θ_d^*) satisfying the following property:

$$V(\theta_g^*, \theta_d) \leq V(\theta_g^*, \theta_d^*) \leq V(\theta_g, \theta_d^*)$$

for all (θ_g, θ_d) in the neighborhood of (θ_g^*, θ_d^*).

We can know $v'(\theta_g, \theta_d)$ is positive define $\Leftrightarrow$ $\nabla^2_{\theta_g} V(\theta_g, \theta_d) \succ 0$ and $\nabla^2_{\theta_d} V(\theta_g, \theta_d) \prec 0$. In Mazumdar et al. (2019) authors show that a local Nash equilibrium (θ_g^*, θ_d^*) means that $v'(\theta_g^*, \theta_d^*)$ is positive define. Conversely if (θ_g^*, θ_d^*) satisfying:

$$v(\theta_g^*, \theta_d^*) = 0 \quad v'(\theta_g^*, \theta_d^*) \succ 0$$

The contribution of their work is they notice that the previous methods just find the solution of the following gradient flow:

$$\dot{\theta} = -v(\theta)$$

However, they don't use the property of the Jacobian matrix of v, which may lead a wrong solution. The authors firstly show the following gradient flow will just find the local Nash equilibrium we want.

$$\dot{\theta} = -h(\theta) = -\frac{1}{2}\left(v(\theta) + v'(\theta)^T(v'(\theta)^T v'(\theta) + \lambda(\theta)I)^{-1} v'(\theta)^T v(\theta)\right)$$

where $\lambda \in \mathcal{C}^2$ is such that $0 \leq \lambda(\theta) \leq \xi$ for all θ and $\xi > 0$ and $\lambda(\theta) = 0 \Leftrightarrow v(\theta) = 0$. I think the construction of such gradient may be from the following intuition:

$$\begin{bmatrix} \theta_g^{k+1} \\ \theta_d^{k+1} \end{bmatrix} = \begin{bmatrix} \theta_g^k \\ \theta_d^k \end{bmatrix} + \gamma v(\theta_g^{k+1}, \theta_d^{k+1})$$

This means we need to look one more step of the other players' action. Use the new gradient flow, we can construct the following two-timescale approximation which can avoid some point which is not local Nash equilibrium.

$$\theta^{k+1} = \theta^k - a^k h_1(\theta^k, \phi^k)$$
$$\phi^{k+1} = \phi^k - b^k h_2(\theta^k, \phi^k)$$

Where h_1 and h_2 defined as following:

$$h_1(\theta, \phi) = \frac{1}{2}(v(\theta) + v'(\theta)^T \phi)$$
$$h_2(\theta, \phi) = v'(\theta)^T v'(\theta)\phi - v'(\theta)^T v(\theta) + \lambda(\theta)\phi$$

and the sequences of step size $\{a^k\}$ and $\{b^k\}$ satisfying $\sum a^k = \infty$, $\sum b^k = \infty$, $\sum (a^k)^2 < \infty$,$\sum (b^k)^2 < \infty$, $\lim_{t\to\infty} \frac{a^k}{b^k} = 0$. (Mazumdar et al., 2019) then shows this discrete method will converge to the same solution of original ODE.

Besides this method, an another method named as Symplectic Gradient Adjustment proposed by Balduzzi et al. (2018) can also do well, in which the authors directly consider n-Player differentiable games and also use the decomposition of the Jacobian matrix to get it.

5.3 Alternative approach with mean-field analysis

It can be seen from previous sections that many analysis relies on gradient flow. Although this simplifies the analysis, it is well known that the properties of gradient flow may not carry over when they are being discretized. Also, we notice that in gradient flow, the idealized training process is always deterministic. Then, as pointed out by Wang et al. (2019), 'the stochasticity and the effect of the noise is essentially ignored, which may not reflect practical situations'. Therefore, rather than analyze in the ccontinuous time limit, an alternative approach is to analyze in the overparameterization limit, an idea borrowed from an emerging area called *mean-field analysis* (Mei et al. (2018)).

5.3.1 A solvable high-dimensional model of GAN

(Wang et al., 2019) proposed the first work we know of that adopted this approach. First, we assume the data are generated by a latent linear model with additive noise:

$$\begin{aligned} \mathbf{y}_k =& U\mathbf{c}_k + \sqrt{\eta_T}\mathbf{a}_k \\ \mathbf{y}_k :& \text{data points} \\ U :& \text{orthogonal matrix (without loss of generality)} \\ \mathbf{c}_k :& \text{latent code drawn from unknown distribution} \\ \sqrt{\eta_T} :& \text{strength of noise} \\ \mathbf{a}_k :& \text{noise} \end{aligned}$$

Then, we analyze a GANs with single-layer, infinitely-wide linear generator and discriminator:

Linear generator for linear data:

$$\begin{aligned}\tilde{\mathbf{y}_k} =& V\tilde{\mathbf{c}_k} + \sqrt{\eta_G}\tilde{\mathbf{a}}_k \\ & \mathbf{c}_k \text{ drawn from fixed distribution}\end{aligned}$$

Single-layer discriminator:

$$\begin{aligned}D(\mathbf{y};\mathbf{w}) =& \hat{D}(\mathbf{y}^T\mathbf{w}) \\ & \hat{D} : \text{arbitrary activation function}\end{aligned}$$

Training objective:

$$\min_V \max_w \mathop{\mathbb{E}}_{\mathbf{y}\sim P(\mathbf{y};U)} \mathop{\mathbb{E}}_{\tilde{\mathbf{y}}\sim \tilde{P}(\tilde{\mathbf{y}},V)} L(\mathbf{y},\tilde{\mathbf{y}};\mathbf{w})$$

$$\begin{aligned}L(\mathbf{y},\tilde{\mathbf{y}};\mathbf{w}) =& F(\hat{D}(\mathbf{y}^T\mathbf{w})) - \tilde{F}(\hat{D}(\tilde{\mathbf{y}}^T\mathbf{w})) - \frac{\lambda}{2}H(\mathbf{w}^T\mathbf{w}) + \frac{\lambda}{2}tr(H(V^TV)) \\ \lambda > 0 \;& F, \tilde{F}, H : \text{ arbitrary element-wise functions}\end{aligned}$$

We further assume noise a_k and $\tilde{a_k}$ are i.i.d. standard Gaussian.

Let's pause for a while and recall that a major problem in GANs training is dimensional misspecification, and a proposed solution is the introduction of additive noise that has support everywhere. Notice this is formation completely avoided the problem with additive Gaussian noise, making it unrealistic and impractical.

Continue on, we define the microscopic and macroscopic states of the training process.

The microscopic state is defined in a straightforward way, as the collection of the parameters:

$$X_k \triangleq [U, V_k, \mathbf{w}_k]$$

The macroscopic state, however, is a bit involved:

$$(P_k, q_k, r_k, S_k, z_k)$$

$$\begin{aligned}P_k \triangleq U^TV_k \quad q_k \triangleq& U^Tw_k \quad r_k \triangleq V_k^Tw_k \\ S_k \triangleq V_k^TV_k \quad & z_k \triangleq w_k^Tw_k\end{aligned}$$

We can gain some intuition into this definition by noticing that every element is a collection of column-wise inner products between the parameters, excluding U^TU because we assume U is orthogonal. Intuitively, the macroscopic state captures how close in angles each pair of columns is among all parameters.

We then formally define the macroscopic state as the shorthand of the previous definition:

$$M_k \triangleq X_k^TX_k = \begin{bmatrix} I & P_k & q_k \\ P_k^T & S_k & r_k \\ q_k^T & r_k^T & z_k \end{bmatrix}$$

We now introduce a potentially very strong assumption:

The initial macroscopic state M_0 satisfies $\mathbb{E}\,||M_0 - M_0^*|| \leq C/\sqrt{n}$, where M_0^* is a deterministic matrix and C is a constant not depending on n.

We shall see the effect of this assumption shortly. Now, with further assumptions, we reach our first conclusion:

Fix $T > 0$, for each $k = \lfloor tn \rfloor$ for some $t \in [0, T]$, the macroscopic state M_k converges to a deterministic number $M(t)$, and the convergence rate is $O(1/\sqrt{n})$.

$$\max_{0 \leq k \leq nT} \mathbb{E}\,||M_k - M(\frac{k}{n})|| \leq \frac{C(T)}{\sqrt{n}}$$

To understand the conclusion, recall that we are analyzing in the overparameterization limit $n \to \infty$. Plug this in, we get:

$$\max_{0 \leq k < \infty} \mathbb{E}\,||M_k - M(0)|| \leq 0$$

Recall the aforementioned assumption, and also plug in $n \to \infty$, we get:

$$\mathbb{E}\,||M_0 - M_0^*|| \leq 0$$

Therefore, we can see in the overparameterization limit, the macroscopic state of the training process is deterministic, and converges to the initial state.

Although this probably can connect to Neural Tangent Kernels (Jacot et al. (2018)), one would naturally regard this conclusion as very weird, and suspect the assumption actually begs the question.

We now state the conclusion on the microscopic state, and the final conclusion of this work:

the microscopic dynamics can be described by an stochastic differential equation. By connecting the macroscopic dynamics and the microscopic dynamics and further assume $\lambda \to \infty$ (strict regularization), we obtain an interesting insight into the effect of noise:

- If the noise is too small, the training process can be trapped in *mode collapse.*
- If the noise is too big, the training process can be trapped in an *oscillating phase.*

This is an interesting insight reminiscent to the informal argument given by (Arjovsky and Bottou, 2017). However, this conclusion comes from very strong assumptions, which is a common issue in mean-field analysis. This work does point to the interesting direction of analyzing in the overparameterization limit, and future works might improve upon these deficiencies.

6. Generalization

As seen in Section 5, when we are busy analyzing convergence to equilibrium, we run the risk of forgetting our original purpose which is to converge to Nash equilibrium. Similarly, when we are analyzing the dynamics of GANs, we should not forget our original purpose is for the generator to generalize. One can then ask the question whether the (Nash) equilibrium actually do generalize well? In the last section, we review a work on the generalization of GANs at equilibrium points (Arora et al. (2017)).

First we point out a framework for analyzing generalization of unsupervised learning is very difficult to construct. The problem is, in supervised learning, generalization is typically captured by the risk:

$$\mathop{\mathbb{E}}_{x \sim p_{data}}[l(f(x;\theta(X)))]$$

where l is the loss function, f is the learned model parameterized by θ, and X is the training set. The difficulty in unsupervised learning is typically no obvious loss function is provided. In particular, in a likelihood-free approach like GANs, a loss function for the whole model is probably nonexistent. We can now discuss the importance of the work by (Arora et al., 2017) where a criteria for the generalization of GANs is proposed by composing the idea of risk minimization in supervised learning and divergence minimization in GANs.

The proposed criteria for generalization is defined as follows:

Let $x_1, \ldots, x_m$ be the training samples. Let $\mathbb{P}_{data}$ be the real distribution of data, $\hat{\mathbb{P}}_{data}$ denote the uniform distribution over $x_1, \ldots, x_m$, $\mathbb{P}_{gen}$ be the distribution of all generated samples, and $\hat{\mathbb{P}}_{gen}$ be the particular distribution of polynomial number of generated samples. Let d be a distance (or divergence) between distributions. We say $\mathbb{P}_{gen}$ generalizes with error ϵ if

$$|d(\mathbb{P}_{data}, \mathbb{P}_{gen}) - d(\hat{\mathbb{P}}_{data}, \hat{\mathbb{P}}_{gen})| \leq \epsilon$$

holds with high probability.

Intuitively, generalization in GANs means that the population distance between the true and generated distribution is close to the empirical distance between the empirical distributions. We want to make the first term small, but we can only optimize towards the second term.

(Arora et al., 2017) then showed that JS-divergence and Wasserstein distance do not generalize with any polynomial number of examples. Therefore, (Arora et al., 2017) proposed neural network distance as an alternative for measuring generalization.

Let $\mathcal{F}$ be a class of functions from $\mathbb{R}^d$ to $[0,1]$ such that if $f \in \mathcal{F}; 1 - f \in \mathcal{F}$. Define the $\mathcal{F}$-divergence w.r.t. ϕ between two distributions of μ and ν as:

$$d_{\mathcal{F},\phi} = \sup\nolimits_{D \in \mathcal{F}} \mathop{\mathbb{E}}_{x \sim \mu}[\phi(D(x))] + \mathop{\mathbb{E}}_{x \sim \nu}[\phi(1 - D(x))] - 2\phi(1/2)$$

Further let $\mathcal{F}$ to be a class of neural nets with a bound n on the number of parameters. We then informally refer to $d_{\mathcal{F}}$ as the neural net distance. It can be proven that although the Wasserstein distance do not generalize, the surrogate loss used in Wasserstein GAN does generalize.

However, the neural network distance does have its own problems. In particular, the neural net distance $d(\mu, \nu)$ can be small even if μ, ν are not very close. Therefore, further investigations are still required in this area.

7. Conclusion

In this survey, we reviewed recent theoretical advances on the dynamics of GANs, highlighting their motivations while also pointing out their deficiencies. We organize our review loosely by their approaches and assumptions, and showed diverse opportunities for new research in this area. We acknowledge this review is by no means exhaustive. We omitted the more recent works from (Domingo-Enrich et al., 2020; Lin et al., 2020; Lei et al., 2019) due to limited time and recent incidents, and hope their work can be partially represented by the presented works. Inclusion of these works is left as future work.

References

Martin Arjovsky and Léon Bottou. Towards principled methods for training Generative Adversarial Networks. pages 474–482, January 2017. URL `http://arxiv.org/abs/1701.04862`. arXiv: 1701.04862.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *34th International Conference on Machine Learning, ICML 2017*, 1:298–321, January 2017. URL `http://arxiv.org/abs/1701.07875`. arXiv: 1701.07875 ISBN: 9781510855144.

Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and Equilibrium in Generative Adversarial Nets (GANs). *arXiv:1703.00573 [cs, stat]*, August 2017. URL `http://arxiv.org/abs/1703.00573`. arXiv: 1703.00573.

David Balduzzi, Sebastien Racaniere, James Martens, Jakob Foerster, Karl Tuyls, and Thore Graepel. The Mechanics of n-Player Differentiable Games. *arXiv:1802.05642 [cs]*, June 2018. URL `http://arxiv.org/abs/1802.05642`. arXiv: 1802.05642.

Carles Domingo-Enrich, Samy Jelassi, Arthur Mensch, Grant Rotskoff, and Joan Bruna. A mean-field analysis of two-player zero-sum games. *arXiv:2002.06277 [cs, math, stat]*, February 2020. URL `http://arxiv.org/abs/2002.06277`. arXiv: 2002.06277.

William Fedus, Mihaela Rosca, Balaji Lakshminarayanan, Andrew M. Dai, Shakir Mohamed, and Ian Goodfellow. Many paths to equilibrium: GANs do not need to decrease a divergence at every step. *arXiv:1710.08446 [cs, stat]*, February 2018. URL `http://arxiv.org/abs/1710.08446`. arXiv: 1710.08446.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv:1406.2661 [cs, stat]*, June 2014. URL `http://arxiv.org/abs/1406.2661`. arXiv: 1406.2661.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. *arXiv:1706.08500 [cs, stat]*, January 2018. URL `http://arxiv.org/abs/1706.08500`. arXiv: 1706.08500.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. *arXiv:1806.07572 [cs, math, stat]*, November 2018. URL `http://arxiv.org/abs/1806.07572`. arXiv: 1806.07572.

Qi Lei, Jason D. Lee, Alexandros G. Dimakis, and Constantinos Daskalakis. SGD Learns One-Layer Networks in WGANs. *arXiv:1910.07030 [cs, stat]*, October 2019. URL `http://arxiv.org/abs/1910.07030`. arXiv: 1910.07030.

Tianyi Lin, Chi Jin, and Michael I. Jordan. On Gradient Descent Ascent for Nonconvex-Concave Minimax Problems. *arXiv:1906.00331 [cs, math, stat]*, February 2020. URL `http://arxiv.org/abs/1906.00331`. arXiv: 1906.00331.

Eric V. Mazumdar, Michael I. Jordan, and S. Shankar Sastry. On Finding Local Nash Equilibria (and Only Local Nash Equilibria) in Zero-Sum Games. *arXiv:1901.00838 [cs, math, stat]*, January 2019. URL `http://arxiv.org/abs/1901.00838`. arXiv: 1901.00838.

Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33): E7665–E7671, August 2018. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1806579115. URL `https://www.pnas.org/content/115/33/E7665`.

Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of GANs. *Advances in Neural Information Processing Systems*, 2017-Decem(Nips):1826–1836, 2017. ISSN 10495258. arXiv: 1705.10461.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? *35th International Conference on Machine Learning, ICML 2018*, 8:5589–5626, 2018. arXiv: 1801.04406 ISBN: 9781510867963.

Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled Generative Adversarial Networks. *Advances in Neural Information Processing Systems*, pages 469–477, November 2016. ISSN 10495258. URL `http://arxiv.org/abs/1611.02163`. arXiv: 1611.02163.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral Normalization for Generative Adversarial Networks. February 2018. URL `https://arxiv.org/abs/1802.05957v1`.

Vaishnavh Nagarajan and J. Zico Kolter. Gradient descent GAN optimization is locally stable. *arXiv:1706.04156 [cs, math, stat]*, January 2018. URL `http://arxiv.org/abs/1706.04156`. arXiv: 1706.04156.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: training generative neural samplers using variational divergence minimization. *arXiv:1606.00709 [cs, stat]*, June 2016. URL `http://arxiv.org/abs/1606.00709`. arXiv: 1606.00709.

Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing Training of Generative Adversarial Networks through Regularization. *arXiv:1705.09367 [cs, stat]*, November 2017. URL `http://arxiv.org/abs/1705.09367`. arXiv: 1705.09367.

Chuang Wang, Hong Hu, and Yue Lu. A Solvable High-Dimensional Model of GAN. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 13782–13791. Curran Associates, Inc., 2019. URL `http://papers.nips.cc/paper/9528-a-solvable-high-dimensional-model-of-gan.pdf`.