Is This Normal? Testing the Ability of GANs to Learn a 1D Parametric Distribution

Daniel S. Lee
University of Arizona
danielslee@email.arizona.edu

Abstract

Generative Adversarial Networks (GAN) have shown an impressive ability to learn and generate realistic samples from complex data distributions. However, notable challenges remain in training and evaluation. In training, it is often difficult to achieve model convergence, especially in the local case where a GAN is already close to the optimal solution. In evaluation, the high dimensionality and unknown form of common distributions (e.g. face images) leads to an incomplete and often heuristic or subjective-based toolbox. In this work, various popular GANs are trained to sample from a 1D standard normal distribution. Through this empirical experiment, it is found that some but not all standard GAN frameworks are able to learn a distribution. In those that are, local convergence is difficult, and regularization techniques are needed to stabilize training and improve the fit and quality of the final generated distribution.

1 Introduction

Generative adversarial networks [1] (GANs) are a class of generative models that seek to implicitly estimate and generate novel samples from a data distribution. In the GAN framework, optimization is based on game theory: a generative model (called the generator) tries to generate realistic data while a discriminative model (the discriminator) learns to distinguish generated samples from real data. Through a minimax game, both models continually try to improve performance relative to each other. The goal of training is to reach the Nash-equilibrium of this game, in which the generator has learned to produce the true distribution [1].

In practice, GANs have achieved state of the art results. They have become the dominant approach in generating realistic samples from complex, real-world distributions, from common datasets of images [2] to domain specific ones such as convergence maps in cosmology [3]. As a result, there is a growing body of research on GANs which seeks to fully leverage the representation learning abilities of deep learning towards generative settings.

However, difficulties remain in GAN optimization. Gradient descent techniques do not always lead to convergence, and training dynamics are not completely understood. On this front, numerous approaches have been used to stabilize training. These include heuristic and theoretical analyses of training dynamics [2, 4], novel regularization strategies [4], and modified architectures and loss functions [5]. Despite these advances, there have also been empirical surveys [4, 6] of these methods which still suggest inconsistency in their ability to converge.

Added to this is the curse of dimensionality. While in theory, GANs are designed to learn a true distribution of data, in practice data distributions (e.g. of images) are high-dimensional and unknown. There is large gap between the theoretical discussions of convergence, which posit training objectives with fixed points that correspond to indistinguishable distributions (e.g. Jensen–Shannon divergence [1], Earth mover's distance [5]), and actual evaluation metrics, which rely on heuristics or practical considerations. Common examples such as the "visual Turing test" [2] evaluate performance through

Code can be found at https://github.com/Danielslee51/GAN-Standard-Normal

subjective eyes as opposed to more fundamental statistical tools or goodness of fit measures. In fact, recent attempts to learn 1-D, parametric distributions with popular techniques [4, 6] have shown discouraging results, inviting further analysis as to whether current strategies can actually approximate a distribution beyond simply generating realistic samples.

The following work presents an empirical study of this question. A variety of GANs with different loss functions and regularization strategies are trained to sample from a standard 1D Gaussian. Two questions are explored: can popular training techniques truly learn a distribution (rather than just find its mode or generate a particular range of samples), and if so, what are their training dynamics?

2 Related Work

The two surveys that have most directly inspired this work are [6] and [4].

In the former, three GAN variations (original GAN, WGAN, and MMD GAN [7]) are trained to sample from simple 1D parametric distributions. For a variety of transformations (e.g. uniform to Gaussian, Gaussian to Gaussian), the MMD GAN seems to learn the correct distribution, while the original and Wasserstein GAN fail. This is the case as well for a 2D parametric distribution and 1D conditional distribution. The failures of the latter two models occur consistently despite sufficient generator capacity and training iterations, suggesting a need to rethink current evaluation metrics (which they perform well on) and further explore the generalization capabilities of popular GANs.

The latter work investigates local convergence, or the ability of a GAN to converge to its Nash equilibrium point when both models are already close. It posits a heuristic explanation of training instabilities: oscillatory behavior in the parameter space, caused by the fact that the discriminator can often have non-zero or increasing slope on the true data distribution. Using an illustrative example of simple, 2-parameter "Dirac-GAN" with known fixed points, the authors show training instabilities for standard GAN, WGAN, and a variety of regularization techniques. While some techniques (to be visited later) result in quick convergence, the original GAN and WGAN never converge (oscillating forever in a ellipse in the 2D parameter space), while the original GAN with its non-saturating loss alternative spirals for long periods before reaching the solution.

This work attempts to be a hybrid of the two above. Using a large subset of the training techniques explored, GANs will be trained and evaluated on a standard normal distribution. The ability to know a true distribution allows for visualization and evaluation via a larger statistical toolkit. This, in summary, will answer two questions: can common GAN frameworks truly learn a distribution (beyond satisfying weaker evaluation metrics), and if so, what does training look like (i.e. is there speedy or oscillatory local convergence)?

3 Towards Improved Training

Due to the practical issues surrounding GAN convergence, a variety of regularization strategies and alternative loss functions have been proposed. In the original, **vanilla GAN**, the generator (G) and discriminator (D) play a minimax game with the following value function:

$$\min_G \max_D \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

Due to the possible saturation of the generator loss in the beginning of training, an **non-saturating** alternative for the generator was proposed.

$$\min_G \mathbb{E}_{z \sim p_z(z)}[-\log D(G(z))]$$

In this formulation, the discriminator remains the same binary classifier.

To address saturation and instability observed in both of the loss functions above, the **Wasserstein GAN (WGAN)** uses the Wasserstein distance in training:

$$\min_G \max_D W(p_{data}, p_g) = \sup_{||f||_L < 1} \mathbb{E}_{x \sim p_{data}(x)}[f(x)] - \mathbb{E}_{z \sim p_z(z)}[f(G(z))]$$

In practice, neural networks are used to approximate $||f||_L < 1$, the set of 1-Lipschitz functions, and the WGAN has been observed to have more stable training dynamics.

Lastly, [4] proposes a zero-centered gradient penalty (**GAN-GP**), which penalizes the the gradients of the discriminator with respect to the sample space. In this technique, the squared norm of the gradients are added to the standard discriminator loss as a regularization term. Generator training is unaffected, and this work will use the non-saturating loss function for its experiment.

$$R = \lambda * \mathbb{E}_{x \sim p_{data}(x)}[||\nabla_x D(x)||^2]$$

Intuitively, this encourages a smoother discriminator function on the true data distribution, helping local convergence by eliminating oscillatory movements around the parameter space. A similar one-centered gradient penalty [8] can be applied to the WGAN to enforce the Lipschitz constraint.

4 Learning a 1D Gaussian

4.1 Training Setup

In this work, GANs are trained to learn a 1D standard normal distribution from 1D uniform noise.

$$z \sim U(0, 1), x \sim N(0, 1)$$

The following four GANs are used: vanilla GAN, non-saturating GAN, WGAN, and GAN-GP. These were chosen based on their popularity (GAN and WGAN), effectiveness in improving local convergence in [4] (GAN-GP), and use in [4, 6]. Notably, the one-centered gradient penalty for WGAN is omitted based on the observation that it performed similarly to WGAN in the setups of [4, 6].

All generator and discriminator are 4-layer, fully-connected networks with 16, 16, 16, and 1 units. Models will be optimized using Adam without momentum with a learning rate of 1e-4 (a result of experimentation with hyperparameter tuning). The generator to discriminator training ratio is fixed at 5:1, based on 1) experimentation and 2) the observation that training the discriminator to convergence is usually problematic due to practical concerns of vanishing gradient, which are mainly present with high-dimensional distributions that the generator is initialized far from. In this 1D case, the generator is initialized such that $G(z), z \sim U(0, 1)$ is not dramatically different from $x \sim N(0, 1)$. Lastly, the gradient penalty term will be fixed at 10 (the value used in [8]).

Each setting is trained for 50,000 iterations with mini-batch size of 256.

4.2 Evaluation

Models will be evaluated quantitatively with Kolmogorov–Smirnov statistic (KS), which compares the target Gaussian cumulative distribution function with the empirical distribution function:

$$KS = \sup_x |\Phi_{Gaussian}(x) - \Phi_{empirical}(x)|$$

This metric will be captured throughout training to give a sense of training dynamics. To evaluate the final generator, this metric will be generated alongside PDF and CDF comparisons with $1 * 10^7$ samples.

4.3 Results

Figure 1 displays the results of training. At a glance, the standard GAN (vanilla and non-saturating) learn a decent approximation of the standard Gaussian, both with very unstable KS statistic values during training. The WGAN seems to only match the mode, while GAN with gradient penalty learns a satisfactory approximation, along with a stable KS statistic throughout training.

These results lead to the following interpretation of the questions posed earlier for each flavor of GAN.

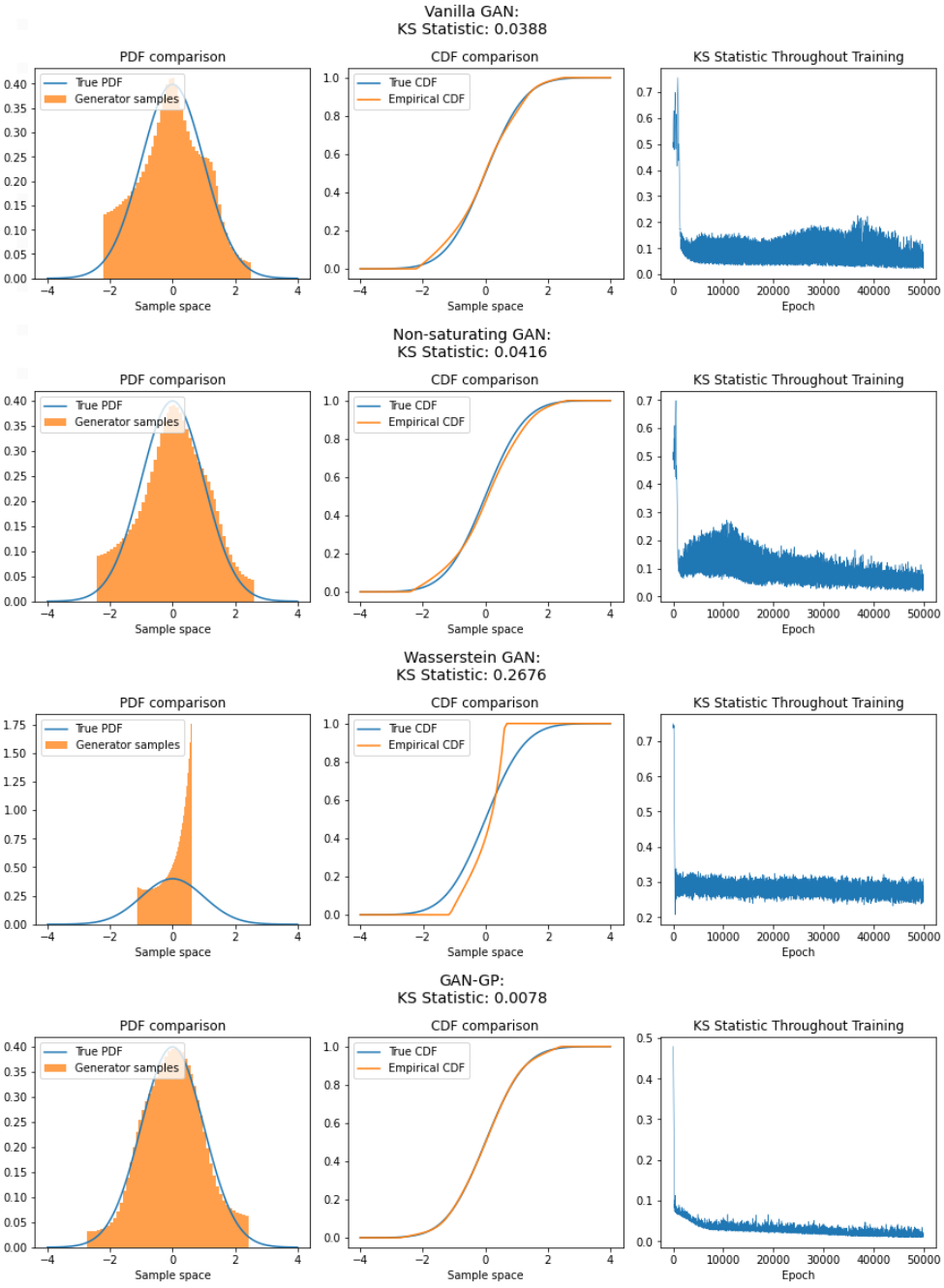

Figure 1: Performance of different GANs on learning a standard normal distribution

Standard GAN (vanilla and non-saturating): In this simplified example, the standard GAN framework seems at least "able" to learn a distribution, in that it has the ability to move from an initialized state to something resembling the distribution. However, practical considerations and local convergence are important to consider. Despite getting quite close, the generator is never able to fill out the edges of the true PDF. In fact, the volatility in KS values implies that the generator was constantly shifting its samples throughout training (as opposed to annealing into a solution), and a manual replay of training dynamics (see the code) confirms this fact. As such, the practical ability to consistently learn a distribution is questionable, likely tied to the issues of local convergence and oscillatory movement explored in [4]. The success of GAN-GP (see below) supports this hypothesis, and suggests that a smoother discriminator function can help reduce such oscillations.

Notably, this contradicts the result of [6], which was not able to train its standard GAN to transform uniform to Gaussian samples. In their experiment, there were differences in the training and model setup, including number of training iterations ($1 * 10^6$), hidden layer size (11, 29, 11, 1) and activation function (ELU). This supports the observation that GANs are extremely sensitive to the details of training. It also suggests that this case study should only be treated as such until more principled and consistent methods are applied or a larger body of experiments is dedicated.

WGAN: Just as in [6], the WGAN fails in this task, only finding the mode of the distribution. Why the WGAN seems to repeatedly fail in simple examples is an interesting question which this work does not attempt to answer. Again, it is important to note that GANs are especially sensitive to training setup and hyperparameter tuning, and that one experiment is not enough to reject a hypothesis.

GAN-GP: A zero-centered gradient penalty applied to the original GAN leads to two satisfying results: an acceptable approximation of the distribution, along with stable training dynamics. The gradient penalty was proposed specifically to improve local convergence, and the smoother discriminator function seems to have had a direct positive effect (note that the non-saturating GAN is identical in every part but the gradient penalty regularizer). In contrast to the standard GAN, the stable KS values imply that the generator was able to "anneal" itself into a solution, capturing the overall bell curve shape early in training and slowly filling out the edges. Another visual replay of the generator throughout training corroborates this. This again confirms the ability of a standard GAN to learn a distribution, and supports the use of regularization to help reduce local oscillations and improve the quality of the final generated distribution.

5 Conclusion

In this work, a small sample of popular GANs are trained to sample from a standard normal distribution. It is found that some but not all GANs demonstrate the ability to learn a distribution, and among those who do, training dynamics in local convergence affect the quality of the final generator. These results support the use regularization techniques that improve the dynamics of local convergence, and suggest further investigation as to why a WGAN has difficulty in this extremely simple problem.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, pages 2672–2680. Curran Associates, Inc., 2014.

[2] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 2234–2242. Curran Associates, Inc., 2016.

[3] Mustafa Mustafa, Deborah Bard, Wahid Bhimji, Zarija Lukić, Rami Al-Rfou, and Jan M. Kratochvil. Cosmogan: creating high-fidelity weak lensing convergence maps using generative adversarial networks. *Computational Astrophysics and Cosmology*, 6(1):1, May 2019.

[4] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? *arXiv:1801.04406 [cs]*, Jul 2018. arXiv: 1801.04406.

[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[6] Manzil Zaheer, Ruslan Salakhutdinov, Barnabás Póczos, and Chun-Liang Li. Gan connoisseur: Can gans learn simple 1d parametric distributions? 2018.

[7] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabas Poczos. Mmd gan: Towards deeper understanding of moment matching network. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 2203–2213. Curran Associates, Inc., 2017.

[8] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5767–5777. Curran Associates, Inc., 2017.