Never Mind the Density, Here's the Level Set

Alexander Sagel
Technische Universität München
a.sagel@tum.de

Martin Gottwald
Technische Universität München
martin.gottwald@tum.de

Abstract

The success of Generative Adversarial Nets is typically explained by their capability to infer the underlying probability distribution from finite sets of data observations and technological enhancements of the original model are driven by the motivation of improving this capability. However, given the poor generalizability of the common probability divergence measures, this explanation appears insufficient. In this work, we argue that of bigger importance than the statistical objective of matching the probability density of the data at hand is the geometrical objective of learning the intrinsic data manifold. We thus propose to design Generative Adversarial Net algorithms following geometric motivations and suggest one possible concept to do so.

1 Introduction

Out of all deep generative models that have emerged in the recent years, the Generative Adversarial Nets (GAN) [3] gained arguably the most attention. An indicator of this is the vast number of GAN flavors that has been introduced. To name a few, we mention the *Least Squares GAN* [9], the *Energy-based GAN* [15], the *Sobolev GAN* [10] and the *Wasserstein GAN* [1]. A common thread running trough the theoretical interpretation justifying the different GAN models is the one of estimating the probability distribution from which the data is assumed to be drawn. For instance, while the optimization of the vanilla GAN can be viewed as the minimization of the *Jensen-Shannen Divergence* (JSD) between the generated and the training data distribution [3], the authors of [1] argue that the *Earth Mover's Distance* (EMD) is a more appropriate probability divergence measure.

The distribution driven motivation puts GANs in the tradition of classical density estimation approaches such as the *Kernel Density Estimator* (KDE) [11] and likelihood based deep generative models such as the *Variational Autoencoder* (VAE) [5]. There is, however, a crucial difference between these approaches and GANs. While the former assume a model for the underlying density, the latter do not. For instance, for a KDE with an approximately band-limited Parzen window to work, the underlying probability density must have low-pass characteristics.

By contrast, GAN models are model free in the sense that they are supposed to estimate the density directly from samples, without making the detour of parametrized model assumptions. However, from this perspective, the GAN training objective is not well defined: On the one hand, a GAN must not simply copy the training samples, but generalize in order to generate new data. On the other hand, if the probability density is not restricted to a particular model, then it is hardly possible to find a reasonable notion of optimality under which the best density approximation of a finite set of samples is not exactly the same set. For a concise explanation of GANs, it is thus necessary to either find an appropriate probability density model or propose an alternative interpretation. In the following, we will pursue the latter approach.

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

2 Density and Geometry

In this section, we will outline some of the shortcomings of viewing GANs as density estimators. For the sake of compactness, we focus on the Wasserstein GANs as a point of reference, due to its popularity and state-of-the-art performance. Let us recall the sample based formulation of the EMD that is used in the batch optimization of the Wasserstein GAN algorithm.

Consider two multisets [6], $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^m$ and $\tilde{\mathcal{X}} = \{\tilde{\boldsymbol{x}}_i\}_{i=1}^m$ that contain training and generated samples, respectively. According to [1], the sample-based EMD is given by

$$D_{\mathrm{EM}}(\mathcal{X}, \tilde{\mathcal{X}}) = \max_{\|f\|_L \leq 1} \frac{1}{m} \sum_{i=1}^{m} f(\boldsymbol{x}_i) - \frac{1}{m} \sum_{i=1}^{m} f(\tilde{\boldsymbol{x}}_i), \tag{1}$$

where $\|\cdot\|_L$ denotes the Lipschitz constant. The term $D_{\mathrm{EM}}(\mathcal{X}, \tilde{\mathcal{X}})$ is minimized, iff

$$\mathcal{X} = \tilde{\mathcal{X}} \tag{2}$$

holds. To see this, first note that $D_{\mathrm{EM}}(\mathcal{X}, \tilde{\mathcal{X}})$ is always non-negative and is therefore minimized at $\mathcal{X} = \tilde{\mathcal{X}}$. Furthermore, if $\mathcal{X} \neq \tilde{\mathcal{X}}$, then we can find a strictly positive lower bound for $D(\mathcal{X}, \tilde{\mathcal{X}})$ by constructing an appropriate 1-Lipschitz window function around a data sample that is contained in $\mathcal{X}$ more often than in $\tilde{\mathcal{X}}$.

In other words, the loss function favors the exact reproduction of the original training data samples. The general experience with (Wasserstein) GANs is the contrary: Their distinguishing feature is their capability to produce morphed versions of the original data [12].

The Wasserstein GAN estimates the EMD by modeling the *critic* function f as a neural network and solving Eq. (1) via restricting the network weights to a compact set. Note that the optimization problem Eq. (1) is constrained to the set of all possible 1-Lipschitz functions. A nearby conclusion is thus that, as long as the critic function is restricted to have the Lipschitz property, in order to assure accuracy of the EMD approximation, the network architecture of the critic must be chosen as general as possible to keep the feasible set of functions as large as possible and therefore as close as possible to the feasible set in Eq. (1). In particular, dense layers should perform at least as good as convolutional ones. However, the experimental results [1] suggest otherwise: It appears that the generation of photo-realistic samples requires a carefully designed convolutional architecture, e.g. the Deep Convolutional GAN discriminator [12], while alternative choices, e.g. the Multilayer Perceptron (MLP), fail. Interestingly, the performance is much less sensitive to the choice of the generator architecture, where replacing the convolutional network by a MLP still leads to reasonable results.

Sensitivity w.r.t. critic architecture choice contradicts the formulation Eq. (1) and thus suggests that the Wasserstein GAN performs better, when it does not directly optimize Eq. (1). This conclusion is corroborated by concerns that have been recently raised regarding the generalization properties of the JSD and the EMD. Specifically, the authors of [2] have shown that estimating the actual JSD or the EMD of two distributions from a finite set of samples is intractable in high-dimensional spaces.

Ultimately, GANs learn a low-dimensional parametrization (the generator) of the data set. This makes them a *manifold learning* [4] algorithm. It is thus reasonable to assume that the capability of a GAN to generalize to unseen data samples is dictated by how well it learns the intrinsic geometry of the underlying data model, rather than its density, provided that, by definition, optimal density estimation is obtained by directly copying the training samples. This assumption is in line with experimental results on toy examples. In particular, when trained on a mixture of 8 two-dimensional Gaussian distributions aligned in a circle, cf. [1], the Wasserstein GAN appears to primarily learn the low dimensional geometry of the circle. The authors explain this behavior by pointing to specific properties of the EMD. However, this explanation might understate the importance of the particular function class the critic belongs to due to its network architecture and, more specifically, the level sets thereof. Their significance was discussed, among others, in [8]. The author considers group action diffeomorphisms such as spatial deformations and suggests that convolutional neural networks (CNN) learn functions that are invariant under these kinds of transformations of the data. According to that, training CNNs on natural images yields functions whose level sets coincide approximately with the sets generated by applying such transformations to an image.

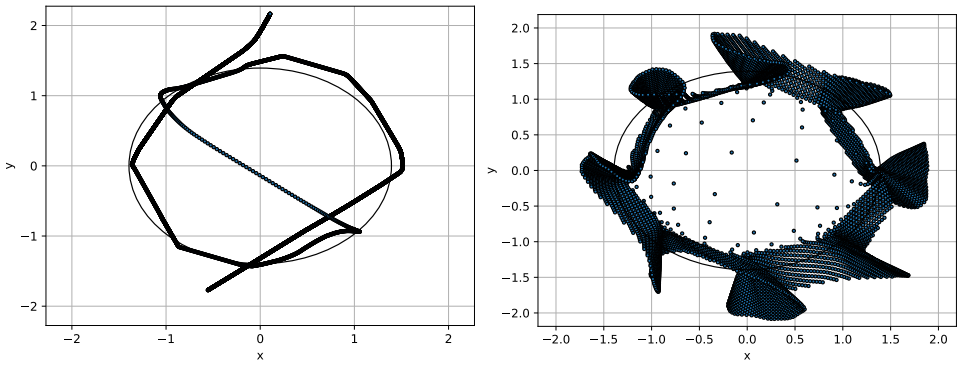

Figure 1: The generator is evaluated on points on a grid. *Left:* Wasserstein GAN is trained with 1-dimensional Gaussian noise. *Right:* Wasserstein GAN is trained with 2-dimensional uniform noise.

Fig. 1 depicts generation results for the 8-Gaussian example. For the sake of visualization, the Generator is evaluated on a regular grid. The Wasserstein GAN is trained for a low dimensional latent representation $\boldsymbol{z}$ with $\dim(\boldsymbol{z}) \in \{1, 2\}$. For $\dim(\boldsymbol{z}) = 2$, it can be observed that the generator preserves the local structure of the grid and contracts it at the centers of the Gaussians.

3 A Level Set View

In order to investigate the manifold learning assumption we formulated in the previous section, we aim at defining an alternative GAN formulation, motivated entirely by geometric heuristics. We propose an approach based on the level sets of the critic function. Consider a function $f : \mathbb{R}^d \to \mathbb{R}$ for which the level set $\mathbb{L} = f^{-1}(0)$ describes the underlying data manifold. Given the outstanding performance of supervised learning via CNNs on different kinds of real-world data, we assume that such a function can be implemented via a CNN.

Consider now a point $\boldsymbol{x} \notin \mathbb{L}$ and a proximity-based projection $\boldsymbol{x}^* = \pi_{\mathbb{L}}\boldsymbol{x}$ onto $\mathbb{L}$. If f is differentiable and $\boldsymbol{x}$ is not too far away from $\mathbb{L}$, then the following approximation holds.

$$f(\boldsymbol{x}) \approx \nabla f(\boldsymbol{x}^*)^\top (\boldsymbol{x} - \boldsymbol{x}^*). \tag{3}$$

Thus, if $f = f_\theta$ is parametrized by a vector θ, then maximizing the magnitude of $f_\theta(\boldsymbol{x})$ w.r.t. θ increases the magnitude of the orthogonal projection of the gradient $\nabla f_\theta(\boldsymbol{x}^*)$ onto $\boldsymbol{x} - \boldsymbol{x}^*$. Meanwhile, if the orthogonal projection of $\nabla f_\theta(\boldsymbol{x}^*)$ onto $\boldsymbol{x} - \boldsymbol{x}^*$ is large, then minimizing the magnitude of $f_\theta(\boldsymbol{x})$ w.r.t. $\boldsymbol{x}$ decreases the norm of $\boldsymbol{x} - \boldsymbol{x}^*$ and thus moves $\boldsymbol{x}$ closer to $\mathbb{L}$.

We thus propose the following loss function for the generator g_ϑ and critic f_θ.

$$\mathcal{L}(\theta, \vartheta) = \mathbb{E}_{\mathbf{x} \sim p_D}[f_\theta(\mathbf{x})^2] - \mathbb{E}_{\mathbf{z} \sim p_S}[f_\theta(g_\vartheta(\mathbf{z}))^2], \tag{4}$$

where p_D refers to the data distribution and p_S to the latent sampling distribution, e.g. standard Gaussian. During optimization of the critic, $\mathcal{L}$ is minimized. As a result, the first term ensures that $f_\theta^{-1}(0)$ is a model for the data manifold, while the second term increases the magnitude of $f_\theta(\boldsymbol{x})$ for points not on the manifold. During optimization of the generator, $\mathcal{L}$ is maximized. The first term has no impact on the optimization and the second term enforces the generated points to move closer to $f_\theta^{-1}(0)$. The optimization is carried out via RMSProp [14] and is summarized in Algorithm 1. To prevent the loss function from exploding, the optimization can be repeated multiple times n_{gen} for each batch. We refer to the described algorithm as *Level Set GAN*.

The Mean Squared Error based objective brings to mind another popular GAN algorithm, the Least Squares GAN. However, the underlying data model of the Level Set GAN is fundamentally different. While the critic optimization of the Least Squares GAN views the real and generated samples as two classes of a classification problem, the critic optimization of the Level Set GAN is more properly described as an anomaly detection, in which the level set serves as a model for the non-anomalous data.

In more formal terms, the relation between level sets and manifolds is the following. Consider a sufficiently smooth, full-rank function $\Gamma : \mathbb{R}^d \to \mathbb{R}^{d-k}$. For any $\boldsymbol{y} \in \mathbb{R}^{d-k}$, the pre-image $\Gamma^{-1}(\boldsymbol{y})$

Algorithm 1: The Level Set GAN

Input: Learning rate α, batch size m, number of generator iterations n_{gen}

Initialize neural network parameters θ, ϑ
while *not converged* **do**
  for *each batch* $\{\boldsymbol{x}_i\}_{i=1}^m$ *of training data* **do**
    Sample $\{\boldsymbol{z}_i\}_{i=1}^m$ from appropriate latent distribution
    $L_\theta \leftarrow \sum_{i=1}^m f_\theta(\boldsymbol{x}_i)^2 - \sum_{i=1}^m f_\theta(g_\vartheta(\boldsymbol{z}_i))^2$
    $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, \nabla_\theta L_\theta)$
    for $t = 1, \ldots, n_{\text{gen}}$ **do**
      Sample $\{\boldsymbol{z}_i\}_{i=1}^m$ from appropriate latent distribution
      $L_\vartheta \leftarrow \sum_{i=1}^m f_\theta(g_\vartheta(\boldsymbol{z}_i))^2$
      $\vartheta \leftarrow \vartheta - \alpha \cdot \text{RMSProp}(\vartheta, \nabla_\vartheta L_\vartheta)$
    end
  end
end

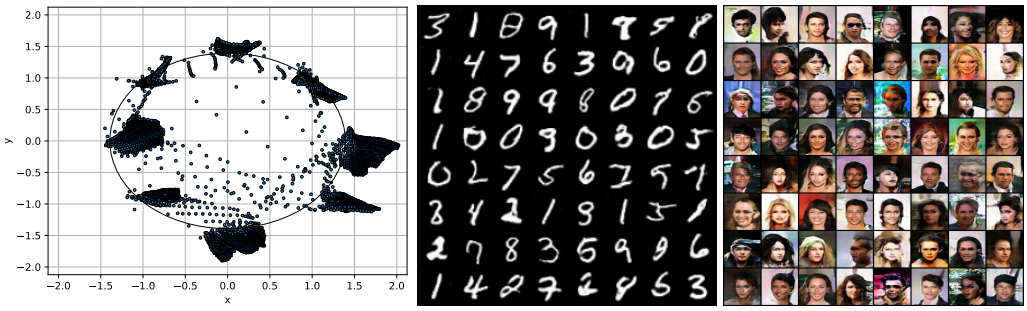

Figure 2: Samples produced by the generator of the Level Set GAN. *Left:* Results for the 8 Gaussian toy problem (trained with uniform noise and evaluated on a grid). *Middle:* Results for MNIST with deep convolutional architecture (trained with Gaussian noise). *Right:* Results for a subset of CelebA with deep convolutional architecture (trained with Gaussian noise)

is a k-dimensional embedded submanifold of $\mathbb{R}^d$. According to this relation, it could be sensible to model the critic as vector-valued, rather than scalar-valued. However, as of now, we have not come up with a reasonable way to control the rank of the critic function, even though the rank of a neural network has attracted considerable attention in recent research, e.g. in [13]. Therefore, we focus on the scalar version for now.

4 Experiments

We observe positive results for the Level Set GAN. Fig. 2 depicts samples generated by training the model on the 8-Gaussian problem, the MNIST dataset and a subset of the CelebA dataset [7], respectively. Even though the quality of the generated images can not yet be considered state-of-the-art, we understand the outcome as a first confirmation that the interpretation of the GAN as manifold learning algorithm is justifiable.

In its current form, the Level Set GAN has issues with exploding values for $\mathcal{L}$ and with mode collapse. The former can be mitigated by increasing the number of stochastic gradient iterations for the generator. The latter needs to be further investigated. Interestingly, replacing the mean squared error formulation in Eq. (4) by an ℓ_1 based loss increases mode collapse. A possible explanation for this is that the ℓ_1 loss favors the scenario, where some parts of the data fit the model very well, while others do not fit the model at all. Mode collapse from a geometrical point of view can be thus described by a scenario, where the manifold overfits to a small subset of the data.

5 Conclusion

In this work, we challenge the commonplace intepretation of GANs as density estimators and propose an alternative view, based on the data geometry and the level sets of the critic function. We suggest to incorporate said view in the design of GAN algorithms and do so by proposing the Level Set GAN. We report proof-of-concept results that give rise to hope that the geometric view is a proper model assumption for GANs. In the future, we aim to improve the Level Set GAN in order to provide a robust and theoretically sound alternative to current GAN based models.

References

[1] 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.

[2] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (GANs). 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 224–232, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[4] Xiaoming Huo, Xuelei (Sherry) Ni, and Andrew K. Smith. *A Survey of Manifold-Based Learning Methods*, pages 691–745.

[5] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[6] Donald Knuth. *The art of computer programming*. Addison-Wesley Pub. Co, Reading, Mass, 1973.

[7] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[8] Stéphane Mallat. Understanding deep convolutional networks. *Phil. Trans. R. Soc. A*, 374(2065):20150203, 2016.

[9] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821, Oct 2017.

[10] Y. Mroueh, C.-L. Li, T. Sercu, A. Raj, and Y. Cheng. Sobolev GAN. *ArXiv e-prints*, November 2017.

[11] Emanuel Parzen. On estimation of a probability density function and mode. *Ann. Math. Statist.*, 33(3):1065–1076, 09 1962.

[12] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. 2015.

[13] Hao Shen. Towards a mathematical understanding of the difficulty in learning with feedforward neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[14] T Tieleman and G Hinton. Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. Technical report, Technical Report. Available online: https://zh. coursera. org/learn/neuralnetworks/lecture/YQHki/rmsprop-divide-the-gradient-by-a-running-average-of-its-recent-magnitude (accessed on 21 April 2017).

[15] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based Generative Adversarial Network. *ArXiv e-prints*, September 2016.