6.883 Science of Deep Learning – Spring 2018 March 7, 2018

Lecture 9: Are GANs Truly Distribution Learners?

Lecturer: Aleksander Mądry *Scribes: Dhroova Aiylam, Shahul Alam, Austin Wang*

1 Recap: Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) have been the focus of much attention recently. The aim of this lecture will be to understand whether GANs really work (in theory and in practice), and how to tell when they don't. Recall that a GAN consists of a generator neural network G and a discriminator network V which are obtained by solving the following minimax optimization problem:

$$\min_G \max_V \mathop{\mathbb{E}}_{x\sim D}[\log(V(x))] + \mathop{\mathbb{E}}_{u\sim\mathcal{N}(0,I)}[\log(1 - V(G(u)))].$$

The first term in the loss samples from the true distribution D; the second term samples from G by seeding it with a random input (e.g. $\mathcal{N}(0, I)$). Viewed as a two-player game, the goal of V is to maximize its discriminative ability, i.e. maximize the expectation of the output on samples from the true distribution and minimize the expectation on samples generated from G on the latent distribution. The goal of G is to model the true distribution as closely as possible, so that V is unable to distinguish between samples from G and samples from D.

There is very little theory to support the fact that GANs can be effectively trained. The original paper [3] makes several assumptions which do not hold in most situations to which GANs are applied. For instance, the authors assume arbitrarily expressive generator and discriminator neural networks and the ability to completely optimize the discriminator network between steps of updating the generator network. Assumptions are also made about the function over which we are optimizing, as the model of a concave-convex game results in certain well-defined convergence properties that are not necessarily generalizable. Nagarajan and Kolter are able to show that [6]:

> Under suitable conditions on the representational powers of the discriminator and the generator, the resulting GAN dynamical system is locally exponentially stable.

That is, if generator and discriminator neural networks (G, V) are initialized sufficiently close to the optimal (G^*, V^*), then training with gradient descent will converge to this equilibrium. However, results of such local nature offer very limited understanding of the global convergence behavior of GANs.

There is no consensus on the best way to train GANs in practice either. Some have suggested that GANs be trained via alternating gradient descent steps on G and V, whereas, others suggest that training should take multiple steps on V, i.e. further optimize the discriminator, for each step on G since the job of the discriminator is "harder." Yet other papers state that in fact V should not be near-optimal, since this kills the gradient in G and can cause the training to settle in a local optima.

To understand better what GANs are actually learning, and if they are learning at all, we turn our attention to analyzing how and in what aspects people have tried to evaluate GANs and to understanding exactly what they produce.

2 Learning Gaussian Distributions with GANs

To begin understanding the dynamics of training and convergence in GANs, Li, Mądry, Peebles, and Schmidt [4] investigate the GAN dynamics in simple cases related to Gaussian distributions, with the goal of finding an example simple enough to mathematically analyze but complex enough to still encapsulate all of the relevant phenomena and potential complications—in particular, vanishing gradients and mode collapse—that arise in GAN convergence.

As Li et. al. find, one can easily prove that when the true distribution is a single univariate Gaussian, convergence of the generator to the true distribution is guaranteed. Hence, they focus mainly on analyzing true distributions constructed as uniform mixtures of two univariate Gaussian distributions with unit weights, showing ultimately that the optimal discriminator always converges, whereas the first order dynamics does not always converge nicely. Formally, they define the generator G to be

$$\mathcal{G} = \left\{ \frac{1}{2}\mathcal{N}(\mu_1, 1) + \frac{1}{2}\mathcal{N}(\mu_2, 1) \mid \mu_1, \mu_2 \in \mathbb{R} \right\},$$

and the discriminator to be

$$\mathcal{D} = \{\mathbb{I}_{[l_1,r_1]} + \mathbb{I}_{[l_2,r_2]} \mid l, r \in \mathbb{R}^2 \mid l_1 \leq r_1 \leq l_2 \leq r_2\},$$

namely the set of indicator functions of sets expressible as two disjoint intervals, a simplification we can make due to the restriction to mixtures of Gaussian distributions and the resulting simplifications in the total variation distance between generators [4]. The optimization task is therefore

$$\hat{\mu} = \arg\min_{\mu} \max_{l,r} L(\mu, l, r), \text{ where}$$
$$L(\mu, l, r) = \mathop{\mathbb{E}}_{x \sim G_{\mu^*}}[D(x)] + \mathop{\mathbb{E}}_{x \sim G_{\mu}}[1 - D(x)].$$

Li et. al. consider two common approaches to solve this optimization problem, *optimal discriminator dynamics* and *first order dynamics*.

Optimal discriminator dynamics involves performing stochastic gradient descent on $G(\hat{\mu}) = \max_{l,r} L(\hat{\mu}, l, r)$. Formally, given an initial $\hat{\mu}^{(0)}$ and step size η_g, we have updates defined as

$$l^{(t)}, r^{(t)} = \arg\max_{l,r} L(\hat{\mu}^{(t)}, l, r),$$
$$\hat{\mu}^{(t+1)} = \hat{\mu}^{(t)} - \eta_g \nabla_{\mu} L(\hat{\mu}^{(t)}, l^{(t)}, r^{(t)}).$$

Because this is difficult to perform in general for more complicated generators and discriminators, one often uses simultaneous gradient iterations on the generator and discriminator, as in *first order dynamics*. Formally, given initial $\hat{\mu}^{(0)}, l^{(0)}, r^{(0)}$ and step sizes η_g, η_d, we have

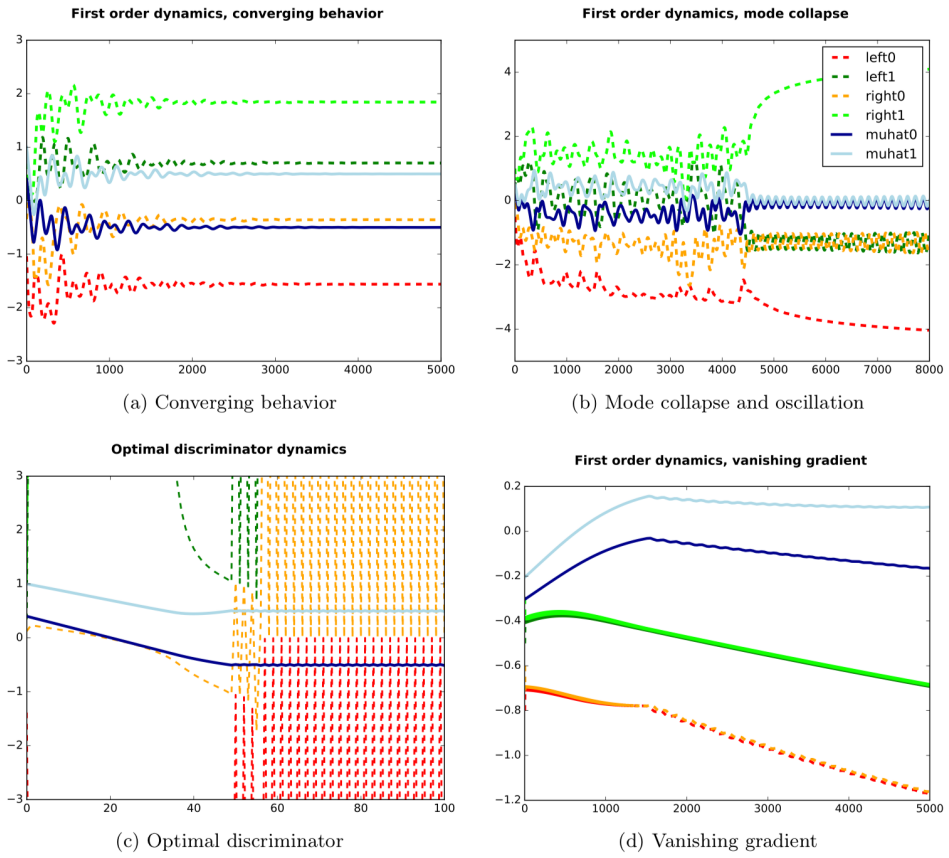
$$\hat{\mu}^{(t+1)} = \hat{\mu}^{(t)} - \eta_g \nabla_{\mu} L(\hat{\mu}^{(t)}, l^{(t)}, r^{(t)})$$
$$r^{(t+1)} = r^{(t)} + \eta_d \nabla_{r} L(\hat{\mu}^{(t)}, l^{(t)}, r^{(t)})$$
$$l^{(t+1)} = l^{(t)} + \eta_d \nabla_{l} L(\hat{\mu}^{(t)}, l^{(t)}, r^{(t)}).$$

In applying each to the mixture of Gaussian distributions, it was found that optimal discriminator dynamics always converged to the true distribution, whereas first order dynamics exhibited a variety of behaviors—either converging, experiencing mode collapse, or converging to a wrong value due to vanishing gradients. Figure 1 shows graphically some of the results of both, in particular demonstrating the different end outcomes that could result from first order dynamics.

We see in figure 1 that, in (a) and (c), the discriminators, or specifically the $\hat{\mu}$ values converge stably to the correct means. On the other hand, more often than not the first order dynamics fail, either when gradients vanish or when modes collapse, i.e. $\hat{\mu}_0$ and $\hat{\mu}_1$ converge on each other, resulting in only one mean represented.

Li et. al. also study a phenomenon they call *discriminator collapse*, in which the local optimization landscape around the current discriminator encourages change in such a way that it loses representational power.

In figure 2 we see an example of discriminator collapse. In each graph, the difference of the true distribution and the generator distribution is given, with regions covered by the discriminator shaded. plot (a) shows the initial configuration of the example, plot (b) shows the result of the globally optimal



(a) Converging behavior

(b) Mode collapse and oscillation

(c) Optimal discriminator

(d) Vanishing gradient

Figure 1: Different GAN behaviors for optimal discriminator dynamics and first order dynamics. The true distribution was G_{μ^*} with $\mu^* = (-0.5, 0.5)$, and the step size was taken to be 0.1. Solid lines represent the coordinates of $\hat{\mu}$, and the dotted lines represent discriminator intervals.

discriminator for the initial condition, and plots (c), (d), and (e) each show the state of the generator and discriminator after 1000 steps, in slightly different variations—the first after just updating the discriminator against a fixed generator, and the other two after applying first order dynamics. From these we see discriminator collapse: the discriminator has incentive to have mass only on regions where the difference between the true distribution and generator distributions is positive, so it risks the collapsing of one of its intervals if in a negative region.

Ultimately, the results of the first order dynamics suggest that they are fundamentally flawed, or at least that any promising implementation of them should somehow be able to resolve the issues of vanishing gradients and mode collapse. The idea of analyzing a restricted set of simple but still interesting distributions helps to give us starting insight into the convergence behavior of GANs at more complicated levels.

3 What Is the Promise of GANs?

Even assuming that training converges to the equilibrium (G^*, V^*), it is not obvious that the result is the model we want. It is natural to hope that G^* has learned a distribution close to the true underlying D, but is this true? And if so, how might one certify it?

The generator G^* determines a "synthetic" distribution $P_S \leftarrow G^*(u)$; consider fixing P_S and solving for the best discriminator V among *all* functions. The loss is

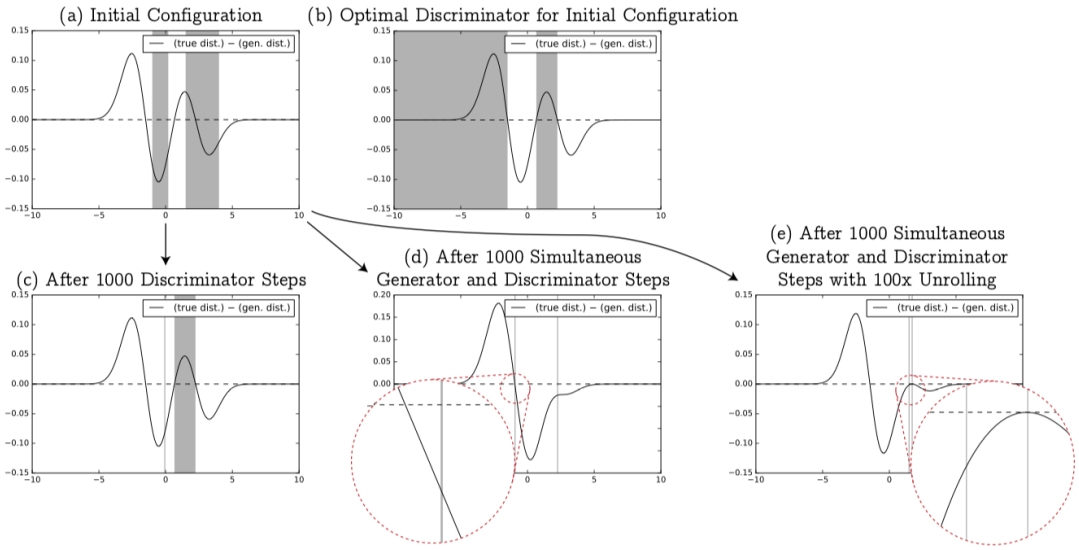


Figure 2: Example of Discriminator Collapse. Initial configuration has $\mu^* = \{-2, 2\}$, $\hat{\mu} = \{-1, 2, 5\}$, left discriminator $[-1, 0.2]$, and right discriminator $[-1, 2.5]$. The step size for plots (c) through (e) was 0.3.

$$\text{loss} = \int_x \Big(P_D(x) \log V(x) + P_S(x) \log(1 - V(x)) \Big) dx$$

where V is non-parametric, so we can simply optimize it pointwise! Setting the derivative w.r.t. V equal to 0 and solving yields

$$V(x) = \frac{P_D(x)}{P_D(x) + P_S(x)}$$

Thus if (G^*, V^*) correspond to a GAN loss of 0,

$$\begin{aligned} 0 &= \int P_D(x) \log \left(\frac{P_D(x)}{P_D(x) + P_{S^*}(x)} \right) + P_{S^*}(x) \log \left(\frac{P_{S^*}(x)}{P_D(x) + P_{S^*}(x)} \right) \\ &= \frac{1}{2} KL \left(P_D || \frac{P_D + P_{S^*}}{2} \right) + \frac{1}{2} KL \left(P_{S^*} || \frac{P_D + P_{S^*}}{2} \right) \\ &= JS(P_D || P_{S^*}), \end{aligned}$$

where KL is the Kullback-Leibler divergence and JS is the Jensen-Shannon divergence.

Hence a GAN (G^*, V^*) that has 0 loss has in fact succeeded in learning a distribution P_{S^*} which is close to P_D in an information-theoretic sense. However, we have assumed that

1. It's possible to find G^*, V^*.
2. The discriminator is infinitely expressive.

The second assumption is quite significant (and perhaps should not be take for granted). Arora, Ge, Liang, Ma, and Zhang [1] observed conversely that a discriminator V with n parameters cannot distinguish between the true distribution P_D and an N-sample approximation $P_{\hat{S}}$ in the sense $\hat{JS}(P_{\hat{S}} || P_D) < \epsilon$ (where $N \approx n \log n / \epsilon^2$). Thus a model with a finite parameterization can be fooled with a finite number of samples – the GAN has learned nothing but a very coarse approximation of P_D. While we would like to be able to prove that training does not settle to such $(\hat{G}, \hat{V})$ in practice, in the next section we will focus on ways to verify this for a GAN that has been trained.

4 Examining a Trained GAN

For fixed generator G producing the synthetic distribution P_S and a true distribution P_D, a number of heuristics are used to tell whether G approximately simulates P_D, i.e. whether G has learned the target distribution. While these heuristics can never verify whether GANs actually accomplish this, they can provide evidence that either bolsters or undermines our confidence in GANs.

One approach is to check the support, i.e. generate samples from P_S and verify that that they are in P_D. Historically this was done by inspection, so if a GAN was trained on a distribution D of faces, its output would be examined to ensure that the results were indeed "face-like." Of course this approach requires manual sample examination and is far from conclusive.

The *inception score* provides a more structured way of doing this. A neural network is first trained on ImageNet and then fed images sampled from the GAN. If $KL(p(y \mid x)||p(y))$, i.e. the KL distance between the distribution of labels $p(y)$ and the softmax probabilities $p(y|x)$, is high, then a high inception score is assigned to the generator. The intuition here is that the distribution of labels should be somewhat uniform, whereas if the output of the GAN is meaningful then the neural network should have a low-entropy belief about the label of any particular image.

One drawback of using the inception score is that it's reasonably easy to fool. Consider a potentially complex distribution with three modes of different labels. A GAN which simply randomizes over the modes will receive a high inception score, but it has clearly failed to learn the target distribution.

Another idea is to show that the GAN has not simply memorized the training data. One could sample from the GAN and measure the closeness of the sample to its nearest neighbor in the training set – hopefully, this will not be too small. However, it is not clear how to define "closeness" . The usual ℓ_2-norm, for instance, is not robust to transformations such as shifts and changing pixel values that have little impact in image-space.

Radford, Metz, and Chintala [7] propose using the latent space embedding to understand the output of GANs. The authors interpolate a seed x from u to u' and study how the output $G(x)$ varies from $G(u)$ to $G(u')$. Rather than obtaining images that were simply a convex combination of the endpoints $G(u)$ and $G(u')$, they found the output was semantically meaningful (à la word2vec). This evidence would seem to support the fact that GANs are actually learning.

Another property of a GAN that has truly learned the target distribution is that its support should not be too small. Arora and Zhang [2] suggest a heuristic based on the birthday paradox: if after sampling s images from the GAN there is a collision, the support of the distribution has size s^2 (assuming a roughly uniform distribution over images). Again the issue of how to measure collision/closeness arises, but with the caveat that diversity is only ever overestimated, never underestimated. Using this heuristic the authors were able to experimentally verify that the diversity of the distribution grew with the number of parameters of the discriminator, as one might expect.

Santurkar, Schmidt, and Mądry [8] studied the diversity of the generated distributions of several popular GANs with a focus on the problem of mode collapse using automated classification-based measures. They consider the Large-Scale CelebFaces Attributes dataset [5] and the Large-Scale Scene Understanding datasets [9], both of which are rich with annotations—for instance, the faces are labeled as male or female and smile or no smile. Their methodology involved training a simple classifier to distinguish attributes based on image input, which proved to do so successfully with high confidence. The intent of such a classifier was to be able to label images sampled from the generator trained on these datasets and so give a measure of the distribution of the generator output, doing all of this in a fairly automated process rather than by manual annotation. While the distribution of the annotations among the four classes was close to uniform for some GANs, for instance ALI, there were many other GANs for which the distribution of annotations were highly asymmetric, which suggests that mode collapse can be a problem in practice (see figure 4).

There are a plethora of other evidence that further encourage this skeptical take on GANs. For example, when the authors compared the spectrum of the covariance matrix of the GAN-generated images to that of the true training data they found that most directions were dropped. In addition, classifiers trained on the GAN distribution generalized more poorly than those trained on the actual images due to serious overfitting. Only the simplest model, a linear classifier, was able to avoid overfitting, but even then it exhibited significant underfitting on the training set (4).

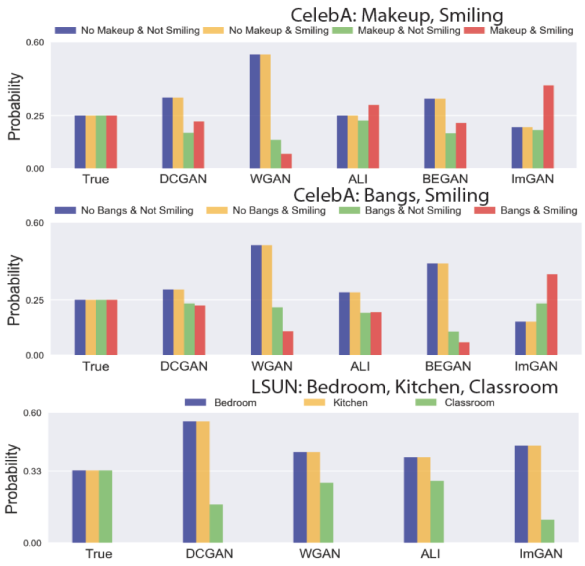


Figure 3: Distributions of images generated from GANs within classification categories, based on trained annotator

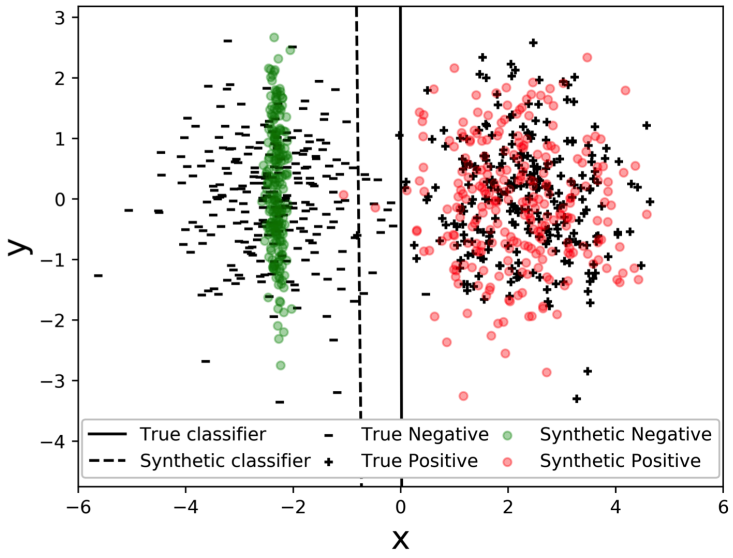


Figure 4: Underfitting of linear classifier on GAN distribution. A simple linear classifier misclassifies much of the original data when applied to the output from a GAN trained on the original data.

Also, another piece of empiric evidence is that the authors found that training with hundreds of thousands of GAN-generated images was just as effective as using a few hundred samples from the actual training data. Taken as a unified corpus, these bits of evidence support the idea that GAN-generated data is derived from a distribution that is much less diverse than the true training distribution. If we were to believe that GANs do indeed learn the underlying distributions that they are trained on, then we would expect that most of the aforementioned heuristics would confirm this, but this has not been the case so far.

References

[1] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). *CoRR*, abs/1703.00573, 2017.

[2] Sanjeev Arora and Yi Zhang. Do gans actually learn the distribution? an empirical study. *CoRR*, abs/1706.08224, 2017.

[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] Jerry Li, Aleksander Madry, John Peebles, and Ludwig Schmidt. Towards understanding the dynamics of generative adversarial networks. *CoRR*, abs/1706.09884, 2017.

[5] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. *CoRR*, abs/1411.7766, 2014.

[6] Vaishnavh Nagarajan and J. Zico Kolter. Gradient descent GAN optimization is locally stable. *CoRR*, abs/1706.04156, 2017.

[7] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

[8] Shibani Santurkar, Ludwig Schmidt, and Aleksander Madry. A classification-based perspective on GAN distributions. *CoRR*, abs/1711.00970, 2017.

[9] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365, 2015.