Overview of GANs

Beau Horenberger

April 29, 2021

Contents

1 The Basics of GANs

1.1 Motivating GANs

The goal of a generative adversarial network (GAN) is to fabricate feasible outcomes of an unknown probability distribution for which we only have samples. For example, consider the set of all possible images of cats. We can sample images from this probability distribution (Search "cat" on Google), but the distribution itself is not directly accessible. We do not have a way to see the set of all possible images of cats and their respective probabilities.

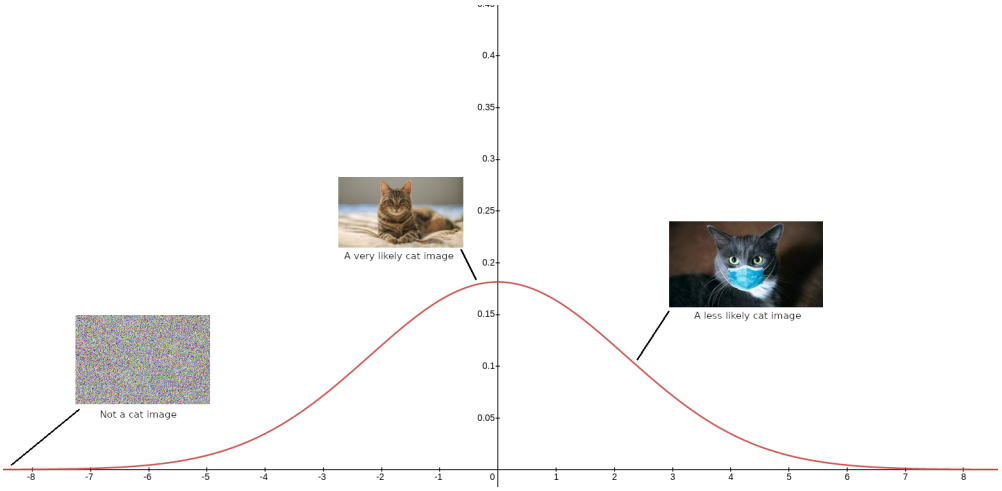

Figure 1: Samples from the unknown probability distribution of cat images

One could train a GAN on sample cat images, and after training, the GAN will generate images which might appear to be images of cats. However, these generated images are fabricated; they do not portray a real cat, and they were not taken by a cat photographer. The GAN has learned to approximate the underlying distribution of cat images, and it fabricates images which are (approximately) convincing images of cats.

GANs actually trains two models simultaneously: a discriminator and a generator. The discriminator learns to discern true samples from fabricated samples, and the generator learns to convert random noise into convincing fabricated samples. At each step of training, the generator provides labeled inputs to the discriminator, and the discriminator provides feedback (assigns loss) to the generator outputs. In this way, the two compete, improving in response to each other. At the end of the training regiment, we are typically most interested in the generator and its properties.

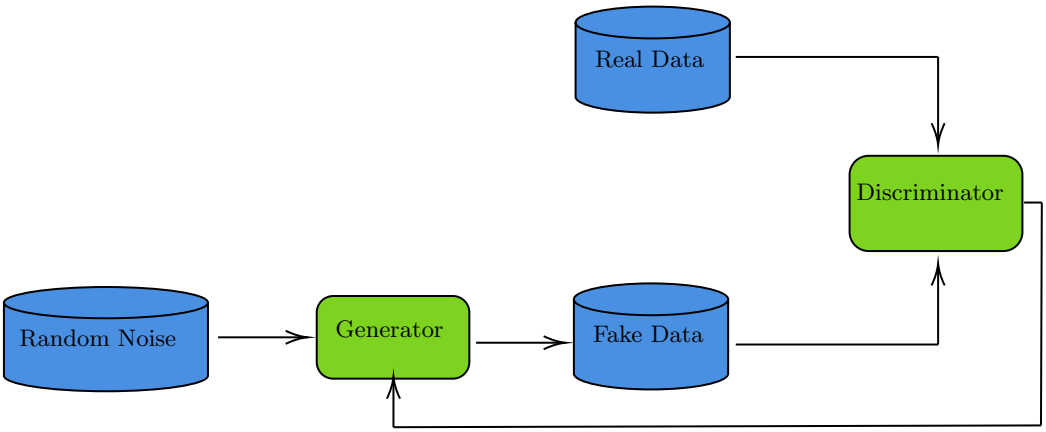

Figure 2: The conceptual structure of a GAN.

1.2 Defining a GAN

GANs were first defined by Goodwell in 2014[3]. This section will largely follow the developments from the original paper. Let's define some basic elements of the theory. Firstly, we define some sets,

$$W = \{\mathbb{R}^n \text{ ; Random noise}\},\ X = \{\mathbb{R}^m \text{ ; Real and fake samples}\},$$
$$Y = [0, 1]$$

Then our discriminator h_D and generator h_G are predictors,

$$h_G : W \to X$$
$$h_D : X \to Y$$

We also introduce random variables $\mathbf{w}$ over W and $\mathbf{x}$ over X, each with distributions $p_{\mathbf{w}}(w)$ and $p_{\mathbf{x}}(x)$, respectively. $\mathbf{w}$ is random noise used to seed the generator, and $\mathbf{x}$ represents the sampling of real samples. We also define $p_G(x)$ as the probability distribution for $h_G(\mathbf{w})$.

We will treat h_D and h_D as multilayer perceptrons using sigmoid cross-entropy loss. In this case, the goal of training both networks simultaneously can be phrased as a minimax problem:

$$\min_{h_G}\max_{h_D} V(h_D, h_G) = \min_{h_G}\max_{h_D}\left[\mathop{\mathbb{E}}_{x\sim\mathbf{x}}[\log h_D(x)] + \mathop{\mathbb{E}}_{w\sim\mathbf{w}}[\log(1 - h_D(h_G(w)))]\right]$$

The first term represents the discriminator's ability to identify true samples. The key property of this design is that the global minimum is achieved if and only if $p_G = p_{\mathbf{x}}$, as we will show next. Further, a simple training algorithm will converge so long as h_G and h_D have enough capacity.[3]

1.3 GANs Minimize Jensen-Shannon Divergence

We can immediately determine the optimal discriminator h_G for a fixed choice of h_G.

Theorem 1. *Let h_G be given. Then the corresponding optimal generator is*

$$h_D^*(x) = \frac{p_{\mathbf{x}}(x)}{p_{\mathbf{x}}(x) + p_G(x)}$$

Proof. We can rephrase $V(h_D, h_G)$ as follows:

$$V(h_D, h_G) = \int_X p_{\mathbf{x}}(x) \log(h_D(X))dx + \int_W p_{\mathbf{w}}(w) \log(1 - h_D(h_G(w)))dw$$

we may combine these and integrate over X since the probability terms will cancel the irrelevant parts of the domain for each term:

$$V(h_D, h_G) = \int_X p_{\mathbf{x}}(x) \log(h_D(x)) + p_G(x) \log(1 - h_D(x))dx$$

Now we need only consider that, so long as $a, b \neq 0$, then

$$\max_{y \in [0,1]} a \log(y) + b \log(1 - y) = \frac{a}{a + b}$$

and thus the proof is complete. □

This means we can rephrase the minimax problem to the equivalent problem,

$$\min_{h_G} \max_{h_D} V(h_D, h_G) = \min_{h_G} \left[\mathop{\mathbb{E}}_{x \sim p_{\mathbf{x}}} \left[\log \frac{p_{\mathbf{x}}(x)}{p_{\mathbf{x}}(x) + p_G(x)} \right] + \mathop{\mathbb{E}}_{x \sim p_G} \left[\log \frac{p_G(x)}{p_{\mathbf{x}}(x) + p_G(x)} \right] \right] =$$
$$\min_{h_G} \left[D_{KL}(p_{\mathbf{x}} \parallel p_{\mathbf{x}} + p_G) + D_{KL}(p_G \parallel p_{\mathbf{x}} + p_G) \right]$$

Where D_{KL} is Kullback-Leibler divergence and D_{JS} is Jensen-Shannon divergence. Now we are prepared to show the key theorem,

Theorem 2. $\min \max V(h_D, h_G)$ *reaches a global minimum with respect to h_G if and only if $p_G = p_{\mathbf{x}}$.*

Proof. First observe that, when $p_G = p_{\mathbf{x}}$, then by Theorem 1, it must be the case that $h_D^* = \frac{1}{2}$. Then we can calculate

$$\max_{h_D} V(h_D, h_G) = \mathop{\mathbb{E}}_{x \sim p_{\mathbf{x}}} \left[\log \frac{1}{2} \right] + \mathop{\mathbb{E}}_{x \sim p_G} \left[\log \frac{1}{2} \right] = -\log(4)$$

Now we will show that this choice of h_G is optimal. Observe that

$$\min_{h_G} \max_{h_D} V(h_D, h_G) = -\log(4) \min_{h_G} \left[D_{KL}(p_{\mathbf{x}} \parallel \frac{p_{\mathbf{x}} + p_G}{2}) + D_{KL}(p_G \parallel \frac{p_{\mathbf{x}} + p_G}{2}) \right] =$$
$$-\log(4) + 2D_{JS}(p_G \parallel p_{\mathbf{x}} + p_G)$$

However, Jensen-Shannon divergence is non-negative and zero only when the distributions are equal, so it follows that we have reached the minimum of $-\log(4)$ when $p_G = p_{\mathbf{x}}$.

□

Goodfellow also demonstrated that a simple algorithm was capable of convergence by iterating gradient descent on the discriminator and then the generator successively. Thus, we have shown the ideal theoretical GAN has the imitative ability we expect.

1.4 Summary

In this introduction, we introduced the generative adversarial network (GAN), which is a combination of two models, a discriminator and a generator. The discriminator intends to discern real samples from fake samples, while the generator fabricates fake samples. Each network is training to outperform the other. When we use neural nets with sigmoid loss, we can describe this interaction as a minimax problem,

$$\min_{h_G}\max_{h_D} V(h_D, h_G) = \min_{h_G}\max_{h_D}\left[\mathop{\mathbb{E}}_{x\sim\mathbf{x}}[\log h_D(x)] + \mathop{\mathbb{E}}_{w\sim\mathbf{w}}[\log(1 - h_D(h_G(w)))]\right]$$

It turns out that the global minimum is achieved when the generator perfectly imitates the sample distribution. We were able to show this by proving that this minimax problem is equivalent to minimizing Jensen-Shannon divergence between the generator's distribution and the sample distribution.

It sounds as if our work here is done, but much remains to be done. Although the theory we've discussed is sound, there are many problems in practice. Namely, convergence often slows to a crawl unless we make modifications to the algorithm. In the next section, we describe these problems in more detail.

2 The Problem With GANs

2.1 Overview

In the previous section, our theory used minimax problems of the form

$$\min_{h_G}\max_{h_D}$$

which implies optimizing h_D for each given h_G. In practice, however, optimizing h_D too much for each h_G within an algorithm causes the performance to drop significantly. Practical algorithms typically only update h_D slightly before updating h_G.

This issue will be the focus of the current section. We will be able to show that, for an arbitrary h_G, the optimal discriminator is typically perfect. We can then show that perfect discriminators cause vanishing gradients, which explains the poor performance observed in naive GAN algorithms.

2.2 Definitions for Distributions and Manifolds

Before we can discuss the existence of perfect discriminators, we need to discuss some terminology. Recall that the generator h_G is a function

$$h_G : W \to X$$

h_G sends random noise from W to fake samples in X. h_G has a corresponding probability distribution, p_G, given by sending the random noise $\mathbf{w}$ through G. In other words, p_G is the distribution for $h_G(\mathbf{w})$. Recall also that we have the probability distribution of true samples given by $p_{\mathbf{x}}$.

When discussing a distribution, let's say $f : A \to B$, we are often only interested in the outcomes $a \in A$ which satisfy $f(a) > 0$. Define the *support* of

a function $f : A \to B$ as the set $\{a \in A : f(a) > 0\}$. We can denote the support as $\operatorname{supp} f$ This is, loosely speaking, the set of "possible" outcomes, but don't say that to a statistician.

It is convenient to think about the support of a probability distribution as being the manifold of possible outcomes. *manifold* intuitively means a collection of shapes or blobs. An n-dimensional manifold can be broken into regions that are homeomorphic to $\mathbb{R}^n$. In other words, the regions can individually described using a continuous function from $\mathbb{R}^n$ with a continuous inverse.

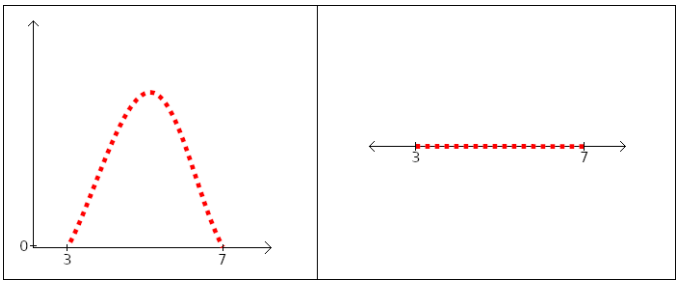

Figure 3: Left: a probability distribution. Right: its support is a line in $\mathbb{R}^1$. A classic one-dimensional manifold.

This way, we can compare $\operatorname{supp} p_G$ to $\operatorname{supp} p_{\mathbf{x}}$ as a means of determining whether the generator produces realistic outcomes. If $\operatorname{supp} p_G$ and $\operatorname{supp} p_{\mathbf{x}}$ have significant overlap, then they have many identical outputs, and consequently a discriminator could never perfectly distinguish the two. In the sections to follow, however, we will show that this circumstance is exceedingly rare, and thus a perfect discriminator usually exists.

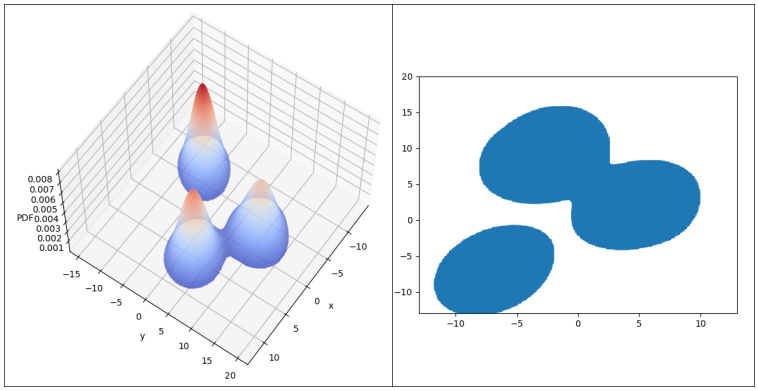

Figure 4: A more interesting distribution and its support. The support is a two-dimensional manifold in $\mathbb{R}^2$

2.3 The Support of p_G Has Low Dimension

The goal of this section is to conceptually clarify the following theorem:

Theorem 3. *Let $h_G : W \to X$ be a function composed by affine transformations and pointwise nonlinearities, which can either be rectifiers, leaky rectifiers, or*

smooth strictly increasing functions (such as the sigmoid, tanh, softplus, etc). Then $h_G(W)$ is contained in a countable union of manifolds of dimension at most $\dim W$*. Therefore, if the dimension of W is less than the one of X, then $h_G(W)$ will be a set of measure 0 in X.*

There is an important consequence of this theorem. Since $\operatorname{supp} p_G \subseteq h_G(W)$ has measure 0 in X, then the manifold $\operatorname{supp} p_G \subseteq X$ has dimension less than X. This is guaranteed by the theorem whenever a standard neural network is used with a low-dimensional noise vector. To visualize what it means for $\operatorname{supp} p_G$ to have low dimension, consider the example distribution in polar coordinates

$$p_{Gex}(r, \theta) = \begin{cases} \frac{1}{2\pi} + \sin \frac{1}{2}\theta, & \text{if } r = 1 \\ 0, & \text{otherwise} \end{cases}$$

The support of p_{circ} is the punctured unit circle, a curve in $\mathbb{R}^2$. In this case,

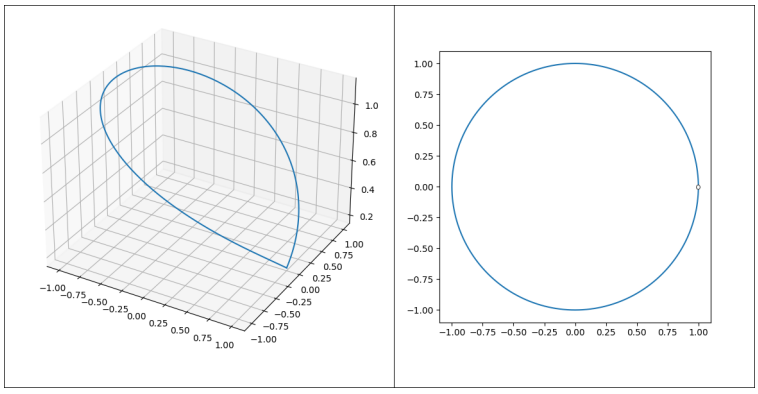

Figure 5: Left: The probability distribution p_{Gex}. Right: its support is the punctured circle $\mathbb{S}^1/\{(1,0)\}$ in $\mathbb{R}^2$. $\dim \mathbb{S}^1/\{(1,0)\} < \dim \mathbb{R}^2$

$\operatorname{supp} p_{Gex} \subseteq X$ has dimension less than X. If the support was instead a solid disk, then it would have dimension equal to X.

Why is this property important? It matters because the intersection of two curves is generally much smaller than the intersection of two solids. If we can assume that both p_G and $p_{\mathbf{x}}$ are lower-dimensional, we will be able to show that it's highly unlikely for them to have significant overlap.

We will not rigorously prove Theorem 3, but we will review some underlying concepts. A mapping $h_G : W \to X$ composed of the functions in lemma 1 can be broken into multiple well-behaved mappings, $h_{G1}, h_{G2}, \ldots h_{Gk}$. These mappings are chosen such that the image $h_G(W)$ will be a subset of $\bigcup_{i=1}^{k} h_{Gi}(W)$. Each mapping is also chosen so that the image $h_{Gi}(W)$ has dimension less than or equal to $\dim W$, and consequently so must their finite union. Therefore, $h_G(W)$ has dimension less than or equal to $\dim W$. Finally, since $\dim W < \dim X$, Sard's theorem implies that the measure of $h_G(W) \subset X$ must be 0.

The tricky part of the proof, which we do not discuss here, is how to pick the mappings h_{Gi} and individually show that $\dim h_{Gi}(W) \leq \dim W$. Some cases simply apply the properties of diffeomorphisms, while others are more involved and require the Morse Lemma.

2.3.1 The Support of $p_{\mathbf{x}}$ is Assumed to Have Low Dimension

Although we will not discuss this in detail here, the manifold hypothesis is the hypothesis that real-world data typically lies on a lower-dimension manifold which is embedded in a higher-dimension space. For example, human faces are typically symmetrical, so we can reduce the dimensionality by only studying the left half of human faces, and very little information is lost. Symmetries such as this are one way in which real-world data appears can be a higher-dimensional embedding of lower-dimensional data.

Figure 6: The dimensionality of facial images is approximately half of the space it is embedded within

We will use this assumption so that both p_G and $p_{\mathbf{x}}$ are now lower-dimensional submanifolds. With this, we are prepared to discuss perfect discriminators.

2.4 Perfect Discriminators Usually Exist

Next we will discuss the existence of perfect discriminators in multiple circumstances. A perfect discriminator $h_D^* : X \to [0, 1]$ satisfies $h_D^*(x) = 0$ whenever $x \in \operatorname{supp} p_G$ and $h_D^*(x) = 1$ whenever $x \in \operatorname{supp} p_{\mathbf{x}}$. Firstly we will show the case where $\operatorname{supp} p_G$ and $\operatorname{supp} p_{\mathbf{x}}$ are disjoint. Here the existence of a perfect discriminator is intuitive, since the outputs from the generator are totally distinct from all real samples. Then, we will discuss the case where $\operatorname{supp} p_G$ and $\operatorname{supp} p_{\mathbf{x}}$ have an intersection. Obviously not every case is distinguishable. As a simple example, suppose we had $\operatorname{supp} p_G = \operatorname{supp} p_{\mathbf{x}}$. Then there is no way a perfect discriminator could exist. However, these cases are exceedingly rare, as we will demonstrate. This is because, since both supports are low-dimension, a random perturbation will (with probability 1) leave only an intersection of measure 0. In turn, this intersection has probability 0 of being observed, so it has no practical effect on the discriminator's performance.

2.4.1 When the Supports of p_G and $p_{\mathbf{x}}$ Are Disjoint

This case is summarized by the following theorem:

Theorem 4. *If distributions p_G and $p_{\mathbf{x}}$ have disjoint compact supports, then there exists a smooth optimal discriminator $h_D^* : X \to [0, 1]$ and $\nabla_x h_D^*(x) = 0$ for any $x \in \operatorname{supp} p_G \cup \operatorname{supp} p_{\mathbf{x}}$ which is a perfect discriminator.*

Proof. Since the supports are compact and disjoint, there is a distance $\delta > 0 = d(\operatorname{supp} p_G, \operatorname{supp} p_{\mathbf{x}})$. We want to create closed disjoint sets containing the

supports, so we define

$$\mathcal{M} = \{x : d(x, \operatorname{supp} p_G) \leq \delta/3\}$$
$$\mathcal{N} = \{x : d(x, \operatorname{supp} p_{\mathbf{x}}) \leq \delta/3\}$$

Since we have closed disjoint sets, it follows from Urysohn's lemma that there exists a function $h_D^* : X \to [0, 1]$ such that h_D^* is a perfect discriminator. Additionally, let $x \in \operatorname{supp} p_G$. Then there exists an open ball $B(x, \delta/3)$ containing x where h_D^* is constant, so it follows that $\nabla_x h_D^*(x) = 0$. The same applies to $\operatorname{supp} p_{\mathbf{x}}$, so the proof is complete. □

The proof leverages the Urysohn lemma, whose underlying concept is simple and uses only intuitive properties of $\mathbb{R}^n$.

2.4.2 When the Supports of p_G and $p_{\mathbf{x}}$ Are Not Disjoint

The discussion of this case will require some additional definitions. In fact, we will discuss the intuition behind the definitions and theorems rather than restating the proofs. To begin, let $\mathcal{M}$ and $\mathcal{N}$ be boundary-free regular submanifolds of $\mathbb{R}^n$. We say that $\mathcal{M}$ and $\mathcal{N}$ *intersect transversally* at x if the tangent spaces at x satisfy $T_x\mathcal{M} + T_x\mathcal{N} = \mathbb{R}^n$. If they do not intersect transversally at x, then we say they *perfectly align* at x.

Suppose $\mathcal{M}$ and $\mathcal{N}$ are curves in $\mathbb{R}^2$ which intersect at x. We could determine whether they intersect transversally by comparing the tangent lines of each curve at x. If the lines are identical, they align perfectly at x. Otherwise, the intersection is transverse. See Figure 6 for an example of both cases.

The most important property of transverse intersections is this:

Theorem 5. *If $\mathcal{M}$ and $\mathcal{N}$ are regular submanifolds of $\mathbb{R}^n$ that don't perfectly align and don't have full dimension, then $\mathcal{M} \cap \mathcal{N}$ is a finite union of manifolds which have lower dimension than both $\mathcal{M}$ and $\mathcal{N}$. Consequently, $\mathcal{M} \cap \mathcal{N}$ has measure 0 in $\mathcal{M}$ and $\mathcal{N}$.*

In essence, this means that whenever two low-dimensional manifolds don't perfectly align, then their intersection is insignificantly small. Perfect alignment is necessary for an intersection to have measure larger than 0. But we cannot yet apply this to $\operatorname{supp} p_G$ and $\operatorname{supp} p_{\mathbf{x}}$, since they may align perfectly. The next theorem gives us a reason to believe that, in practice, the two will never perfectly align.

Theorem 6. *Let $\mathcal{M}$ and $\mathcal{N}$ be regular submanifolds of $\mathbb{R}^n$ that don't have full dimension, and let η and η' be arbitrary independent continuous random variables. Then, with probability 1, $\mathcal{M} + \eta$ and $\mathcal{N} + \eta'$ do not perfectly align.*

This theorem is our grounds for assuming that $\operatorname{supp} p_G$ and $\operatorname{supp} p_{\mathbf{x}}$ do not perfectly align. If they were randomly perturbed at all, then it's guaranteed that alignment would be destroyed. By making this assumption, we can feel the full weight of the following perfect discrimination theorem:

Theorem 7. *Let p_G and $p_{\mathbf{x}}$ be two distributions whose supports are contained in two closed manifolds which do not perfectly align and don't have full dimension. Denote the respective enclosing manifolds $\mathcal{M}$ and $\mathcal{N}$. Assume that p_G and $p_{\mathbf{x}}$ are*

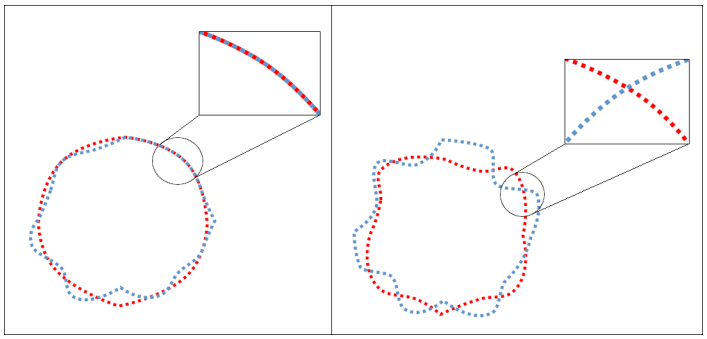

Figure 7: Left: Two manifolds perfectly align on a small segment. Right: A random perturbation has destroyed perfect alignment, leaving only transverse intersections.

continuous (i.e. any set with measure 0 has probability 0). Then for almost all points, there exists an optimal discriminator $h_D^* : X \to [0,1]$ *which is perfect. Additionally, for almost any* x *in* $\mathcal{M}$ *or* $\mathcal{N}$, h_D^* *is smooth in a neighborhood around* x *and* $\nabla_x h_G^*(x) = 0$.

The proof of this theorem begins by excluding the intersection $\operatorname{supp} p_G \cap \operatorname{supp} p_{\mathbf{x}}$ from the supports, which is allowed since the intersection has measure zero. It uses Urysohn's lemma similarly to the disjoint case to create a perfect discriminator. Then, it uses the property of closed sets to create balls around every point in the supports, and it uses this in conjunction with the fact that h_D^* is constant to show that the gradient is zero.

So in summary, we've been able to demonstrate that under reasonable assumptions (manifold hypothesis, noise in p_G and $p_{\mathbf{x}}$), a perfect discriminator exists with probability one. Finally, we will briefly discuss the consequences of this fact.

2.5 Perfect Discriminators Cause Vanishing Gradients

Now that we know the best possible discriminator is always perfect, we can conclude in addition that an improving discriminator causes gradients to vanish.

Theorem 8. *Let* p_G *and* $p_{\mathbf{x}}$ *be two distributions whose supports are contained in two closed manifolds which do not perfectly align and don't have full dimension. Assume that* p_G *and* $p_{\mathbf{x}}$ *are continuous. Let* h_D *be a differentiable discriminator. If* $\sup_{x \in X} |h_D(x)| + ||\nabla_x h_D(x)||_2 < \epsilon$ *and* $\mathbb{E}_{w \sim p_{\mathbf{w}}}[||Jh_G(w)||_2^2] \leq M^2$, *then*

$$||\nabla_{h_G(w)} \mathop{\mathbb{E}}_{w \sim p_{\mathbf{w}}}[\log(1 - h_D(h_G(w)))]||_2 < M \frac{\epsilon}{1 - \epsilon}$$

Consequently,

$$\lim_{\sup_{x \in X} |h_D(x)| + ||\nabla_x h_D(x)||_2 \to 0} ||\nabla_{h_G(w)} \mathop{\mathbb{E}}_{w \sim p_{\mathbf{w}}}[\log(1 - h_D(h_G(w)))]||_2 = 0$$

Note that J indicates the Jacobian matrix. The proof proceeds Jensen's inequality and the chain rule. If, instead of $\nabla_{h_G(w)} \mathbb{E}_{w \sim p_{\mathbf{w}}}[\log(1 - h_D(h_G(w)))]$,

we updated the gradient using $\nabla_{h_G(w)} \mathbb{E}_{w \sim p_{\mathbf{w}}}[-\log(h_D(h_G(w)))]$, it could be shown similarly the the gradient explodes.

Thus, we have constructed a convincing argument for the underlying instability of GANs. Now that the problem is understood theoretically, many attempts have been made to improve upon GANs. We will briefly discuss some of these attempts in the sections to come.

3 GAN Implementations

Here we will exhibit a few variations on the GAN model. We will describe their distinguishing features and demonstrate their implementation on some task.

3.1 GAN With Sigmoid Loss

This is the classic GAN described above, so we need not elaborate on the theory. We simply use the loss function

$$\log(1 - h_D(h_G(w)))$$

between two standard neural networks. The source code can be found in the citations. We used the standard GAN to imitate handwritten numerical digits

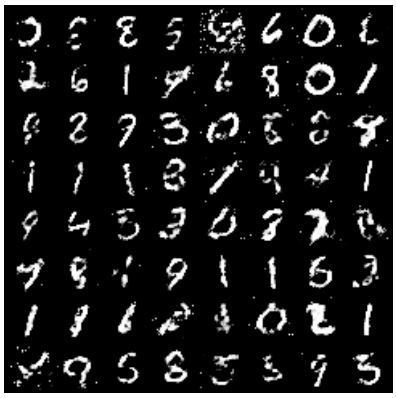

Figure 8: A random sample of GAN outputs after 200 epochs.

using the MNIST database.

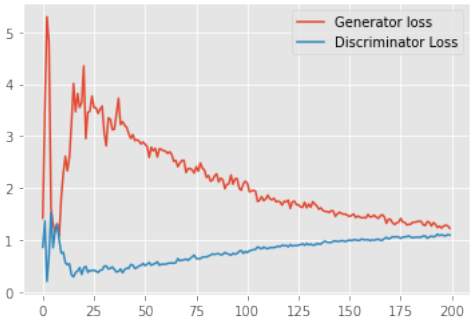

Figure 9: Losses during training for the GAN discriminator and generator networks

3.2 GAN With Least Squares Loss

This modification is an attempt to resolve the problems with GANs by using least squares loss in place of sigmoid loss. Although the first paper to describe this method did not mathematically prove that it overcomes the problem of vanishing gradients, LSGANs perform very well empirically. We again trained

Figure 10: A random sample of LSGAN outputs after 200 epochs.

using the MNIST database for the same number of epochs in order to compare with the standard GAN. Although more investigation is needed, the losses seem substantially lower in the LSGAN training. It's an interesting fact that LSGANs minimize Pearson χ^2 divergence, rather than Jensen-Shannon divergence. Indeed, GANs can be designed to minimize a variety of f-divergences.

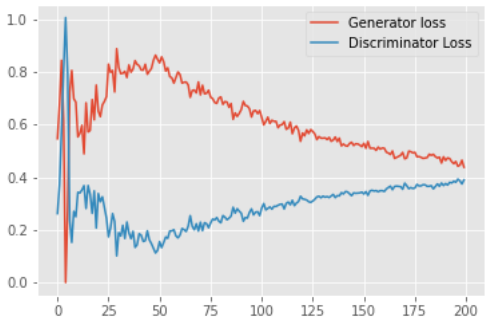

Figure 11: Losses during training for the LSGAN discriminator and generator networks

3.3 Deep Convolutional GAN

The deep convolutional GAN (DCGAN) is a significant modification in which convolutional layers are added to the GAN networks. Convolutional filters are commonly used in imagery-based tasks, so their integration with GANs for image fabrication is somewhat intuitive.

Figure 12: A random sample of DCGAN outputs after 200 epochs.

It's worth noting that the loss clearly behaved differently for the DCGAN over the same quantity of epochs. Subjectively, the DCGAN produced convincing results much earlier than both the GAN and LSGAN. The generator may have approximately converged on its optimal solution, with some noise. Then the discriminator continued to slowly converge on the perfect discriminator. It is an interesting question whether the losses will reach an equilibrium or if the generator loss will diverge. Additionally, will this behavior change if noise is added to the real sample images?

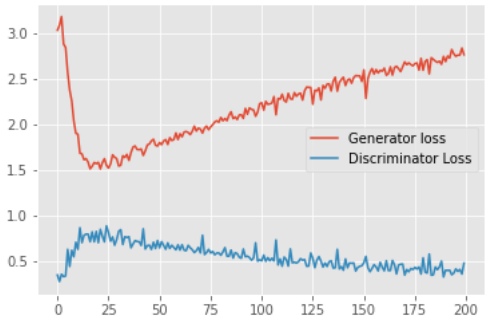

Figure 13: Losses during training for the DCGAN discriminator and generator networks

4 Conclusion

In this paper, we have discussed discussed the concept of GANs and a theoretical model to explain their empirical behavior. We found that GANs are well-suited for generative tasks, but that measures must be taken to address inherent issues with vanishing or exploding gradients. We also discussed variations on the standard GAN model and experimentally observed their performance on the MNIST database.

There are many topics which were not covered in this paper. Other works have created modified GANs to minimize a variety of f-divergences. Wasserstein GANs introduce Earth-Mover distance, which represents the minimum cost of deforming one distribution into the other. These concepts extend into other fields, such as reinforcement learning. However, this paper has completed its goal of introducing the reader to generative adversarial networks and their theory.

References

[1] Arjovsky et al. Towards principled methods for training generative adversarial networks. https://arxiv.org/abs/1701.04862. Accessed: 4/2/2021.

[2] Arjovsky et al. Wasserstein gan. https://arxiv.org/abs/1701.07875. Accessed: 4/2/2021.

[3] Goodfellow et al. Generative adversarial nets. https://papers.nips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf. Accessed: 4/2/2021.

[4] Mao et al. Least squares generative adversarial networks. https://openaccess.thecvf.com/content_iccv_2017/html/Mao_Least_Squares_Generative_ICCV_2017_paper.html. Accessed: 4/2/2021.

[5] Ferlix. Cat-faces-dataset. https://github.com/Ferlix/Cat-faces-dataset. Accessed: 4/2/2021.

[6] Nathan Inkawhich. Dcgan tutorial. https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html. Accessed: 4/2/2021.

[7] Sovit Ranjan Rath. Gan source. https://debuggercafe.com/generating-mnist-digit-images-using-vanilla-gan-with-pytorch/. Accessed: 4/2/2021.