Towards a testable notion of generalisation for generative adversarial networks

Robert Cornish & Frank Wood
Department of Engineering Science, University of Oxford
{rcornish,fwood}@robots.ox.ac.uk

Hongseok Yang
School of Computing, KAIST, South Korea
hongseok00@gmail.com

Abstract

We consider the question of how to assess generative adversarial networks, in particular with respect to whether or not they generalise beyond memorising the training data. We propose a simple procedure for assessing generative adversarial network performance based on a principled consideration of what the actual goal of generalisation is. Our approach involves using a test set to estimate the Wasserstein distance between the generative distribution produced by our procedure, and the underlying data distribution. We use this procedure to assess the performance of several modern generative adversarial network architectures. We find that this procedure is sensitive to the choice of ground metric on the underlying data space, and suggest a choice of ground metric that substantially improves performance.

1 Introduction

Generative adversarial networks (GANs) [Goodfellow et al., 2014] have attracted significant interest as a means for generative modelling. However, recently concerns have been raised about their ability to generalise from training data and their capacity to overfit [Arora and Zhang, 2017, Arora et al., 2017]. Moreover, techniques for evaluating the quality of GAN output are either ad hoc, lack theoretical rigor, or are not suitably objective – often times "visual inspection" of samples is the main tool of choice for the practitioner. More fundamentally, it is sometimes unclear exactly what we want a GAN to do: what is the learning task that we are trying to achieve?

In this paper, we provide a simple formulation of the GAN training framework, which consists of using a finite dataset to estimate an underlying data distribution. The quality of GAN output is measured precisely in terms of a statistical distance D between the estimated and true distribution. Within this context, we propose an intuitive notion of what it means for a GAN to generalise.

We also show how our notion of performance can be measured empirically for any GAN architecture when D is chosen to be a Wasserstein distance, which – unlike other methods such as the inception score [Salimans et al., 2016] – requires no density assumptions about the data-generating distribution. We investigate this choice of D empirically, finding that its performance is heavily dependent on the choice of ground metric underlying the Wasserstein distribution. We suggest a novel choice of ground metric that we show performs well, and also discuss how we might otherwise use this observation to improve the design of Wasserstein GANs (WGANs) [Arjovsky et al., 2017].

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

2 The objective of generative modelling

GANs promise a means for learning complex probability distributions in an unsupervised fashion. In order to assess their effectiveness, we must first define precisely what we mean by this. We seek to do so in this section, presenting a formulation of the broader goal of generative modelling that we believe is widely compatible with much present work in this area. We also provide a natural notion of generalisation that arises in our framework.

Our setup consists of the following components. We assume some distribution π on a set $\mathcal{X}$. For instance, $\mathcal{X}$ may be the set of 32x32 colour images, and π the distribution from which the CIFAR-10 dataset was sampled. We assume that π is completely intractable: we do not know its density (or even if it has one), and we do not have a procedure to draw novel samples from it. However, we do suppose that we have a fixed dataset X consisting of samples $x_1, \cdots, x_{|X|} \overset{\text{iid}}{\sim} \pi$. Equivalently, we have the empirical distribution

$$\hat{X} := \frac{1}{|X|} \sum_{n=1}^{|X|} \delta_{x_n},$$

where δ denotes the Dirac mass.

Let $\mathcal{P}(\mathcal{X})$ denote the set of probability distributions on $\mathcal{X}$. Our aim is to use X to produce a distribution in $\mathcal{P}(\mathcal{X})$ that is as "close" as possible to π. We choose to measure closeness in terms of a function $D : \mathcal{P}(\mathcal{X}) \times \mathcal{P}(\mathcal{X}) \to \mathbb{R}$. Usually D will be chosen to be a *statistical divergence*, which means that $D(P, Q) \geq 0$ for all P and Q, with equality iff $P = Q$. The task of a learning algorithm α in this context is then as follows:

$$\boxed{\text{Select } \alpha(X) \in \mathcal{P}(\mathcal{X}) \text{ such that } D(\alpha(X), \pi) \text{ is as small as possible.}} \tag{1}$$

We believe (1) constitutes an intuitive and useful formulation of the problem of generative modelling that is largely in keeping with present research efforts.

Now, we can immediately see that one possibility is simply to choose $\alpha(X) := \hat{X}$. Moreover, in the case that D is a metric for the weak topology on $\mathcal{P}(\mathcal{X})$ such as a Wasserstein distance, we have that $D(\hat{X}, \pi) \to 0$ almost surely as $|X| \to \infty$, so that, assuming $|X|$ is large enough, we can already expect $D(\hat{X}, \pi)$ to be small. This suggests a natural notion of generalisation: we can say that a choice of α generalises for a given X if

$$D(\alpha(X), \pi) < D(\hat{X}, \pi). \tag{2}$$

In other words, using α here has actually achieved something: perhaps through some process of smoothing or interpolation, it has injected additional information into $\hat{X}$ that has moved us closer to π than we were *a priori*.

3 Generalisation in GANs

The previous section presented (1) as a general goal of generative modelling. In this section, we turn specifically to GANs. We begin by providing a general model for how many of the existing varieties of GAN operate, at least ideally. We then show how this model fits into our framework above, before considering the issue of generalisation in this context.

Most GAN algorithms in widespread use adhere to the following template: they take as input a distribution P, from which we assume we can sample, and compute (or approximate)

$$\Gamma(P) := \arg\min_{Q \in \mathcal{Q}} D_\Gamma(P, Q)$$

for some choices of $\mathcal{Q} \subseteq \mathcal{P}(\mathcal{X})$ and $D_\Gamma : \mathcal{P}(\mathcal{X}) \times \mathcal{P}(\mathcal{X}) \to \mathbb{R}$. In other words, in the ideal case, a GAN maps P to its D_Γ-projection onto $\mathcal{Q}$. Note that we will not necessarily have that $D_\Gamma = D$: D_Γ is fixed given a particular GAN architecture, whereas the choice of D is simply a feature of our problem definition (1) and is essentially ours to make.

In practice, $\mathcal{Q}$ is the set of pushforward measures $\nu \circ G^{-1}$ obtained from a fixed noise distribution ν on a noise space $\mathcal{Z}$ and some set $\mathcal{G}$ of functions $G : \mathcal{Z} \to \mathcal{X}$. Precisely, $\mathcal{Q} = \left\{\nu \circ G^{-1} : G \in \mathcal{G}\right\}$. $\mathcal{G}$

itself usually corresponds to the set of functions realisable by some neural network architecture, and ν is some multivariate uniform or Gaussian distribution. However, numerous choices of D_Γ have been proposed: the original GAN formulation [Goodfellow et al., 2014] took D_Γ to be the Jenson-Shannon divergence, whereas the f-GAN [Nowozin et al., 2016] generalised this to arbitrary f-divergences, and the Wasserstein GAN [Arjovsky et al., 2017] advocated the Wasserstein distance.

In terms of our framework in the previous section, using a GAN Γ amounts to choosing

$$\alpha(X) := \Gamma(\hat{X}) = \arg\min_{Q \in \mathcal{Q}} D_\Gamma(\hat{X}, Q).$$

We emphasise again the important distinction between D and D_Γ. In our setup, minimising D defines our ultimate goal, whereas minimising D_Γ (over $\mathcal{Q}$) defines how we will attempt to achieve that goal. Even if $D \neq D_\Gamma$, it is still at least conceivable that $D(\Gamma(\hat{X}), \pi)$ might be small, and therefore this choice of α might be sensible. Also note that, crucially, Γ receives $\hat{X}$ as input rather than π itself. We only have access to a fixed number of CIFAR-10 samples, for example, not an infinite stream. Moreover, training GANs usually involves making many passes over the same dataset, so that, in effect, sampling from P will repeatedly yield the same data points. We would not expect this to occur with nonzero probability if $P = \pi$ for most π of interest.

The observation that P is $\hat{X}$ rather than π was also recently made by Arora et al. [2017]. The authors argue that this introduces a problem for the ability of GANs to generalise, since, if D_Γ is a divergence (which is almost always the case), and if $\mathcal{Q}$ is too big (in particular, if it is big enough that $\hat{X} \in \mathcal{Q}$), then we trivially have that $\Gamma(\hat{X}) = \hat{X}$. However, in practice, $\mathcal{Q}$ is heavily restricted, since $\mathcal{G}$ is restricted via a choice of neural network architecture; hence we do not know *a priori* whether $\Gamma(\hat{X}) = \hat{X}$ is even possible. As such, we do not see the choice of $\alpha(X) = \Gamma(\hat{X})$ as necessarily a bad idea, and believe that it is an open empirical question as to how well GANs perform the task (1).

4 Testing GANs

Our goal in this section is to assess how well GANs achieve (1) by estimating $D(\alpha(X), \pi)$ for $\alpha(X) = \Gamma(\hat{X})$. This raises some difficulties, given that π is intractable. Our approach is to take D to be the first Wasserstein distance $W_{d_\mathcal{X}}$, where $d_\mathcal{X}$ is a metric on $\mathcal{X}$ referred to as the *ground metric*. This is appealing since, if $(\mathcal{X}, d_\mathcal{X})$ is compact, then $W_{d_\mathcal{X}}$ metricises weak convergence for $\mathcal{P}(\mathcal{X})$ [Villani, 2008]. As a result, if we have a set of samples A from $\alpha(X)$ and a set of samples Y (separate from X) from π, with corresponding empirical distributions $\hat{A}$ and $\hat{Y}$, then $D(\hat{A}, \hat{Y}) \to D(\alpha(X), \pi)$ almost surely as $\min\{|A|, |Y|\} \to \infty$.

As such, to estimate $D(\alpha(X), \pi)$, for $D = W_{d_\mathcal{X}}$, we propose the following. Before training, we move some of our samples from X into a testing set Y. We next train our GAN on X, obtaining $\alpha(X)$. We then take samples A from $\alpha(X)$, and obtain the estimate

$$W_{d_\mathcal{X}}(\hat{A}, \hat{Y}) \approx W_{d_\mathcal{X}}(\alpha(X), \pi),$$

where the left-hand side can be computed exactly by solving a linear program since both $\hat{A}$ and $\hat{Y}$ are discrete [Villani, 2003]. We can also use the same methodology to estimate $W_{d_\mathcal{X}}(\hat{X}, \pi)$ by $W_{d_\mathcal{X}}(\hat{X}, \hat{Y})$, which suggests that a proxy for (2) is given by testing whether

$$W_{d_\mathcal{X}}(\hat{A}, \hat{Y}) < W_{d_\mathcal{X}}(\hat{X}, \hat{Y}). \tag{3}$$

We applied our methodology to test a DCGAN [Radford et al., 2015] applied to the CIFAR-10 dataset. In all cases when computing the relevant Wasserstein distances, our empirical distributions consisted of 10000 samples. Initially we chose the L^2 distance as our ground metric. Figure 1 shows the trajectory of $W_{L^2}(\hat{G}, \hat{Y})$ over training. Strangely, we observe that $W_{L^2}(\hat{G}, \hat{Y}) < W_{L^2}(\hat{X}, \hat{Y})$ very early on in training – at around batch 500 – when the visual quality of samples is very low. This raises some obvious concerns about the appropriateness of W_{L^2} as a metric for GAN quality.

To understand this behaviour, we explored the effect on $W_{L^p}(\hat{X}, \hat{Y})$ of blurring the CIFAR-10 training set X. In particular, we let X and Y each consist of 10000 distinct CIFAR-10 samples in X, and then independently convolved each channel of each image with a Gaussian kernel having

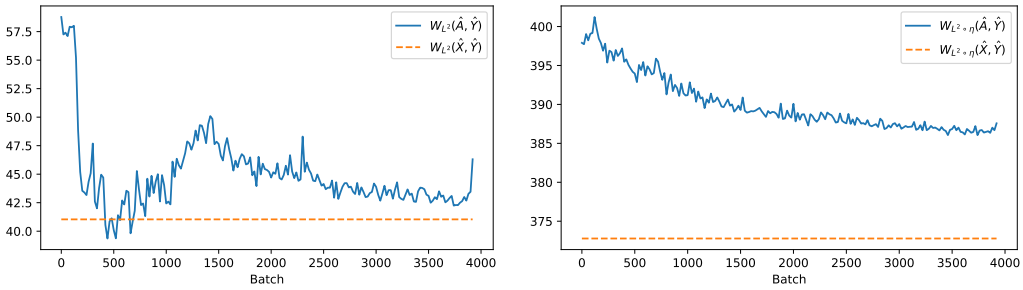

Figure 1: $W_{d_\mathcal{X}}(\hat{G}, \hat{Y})$ over training run for DCGAN trained on CIFAR-10, for $d_\mathcal{X} = L^2$ (left) and $d_\mathcal{X} = L^2 \circ \eta$ (right). $W_{d_\mathcal{X}}(\hat{X}, \hat{Y})$ is shown dashed.

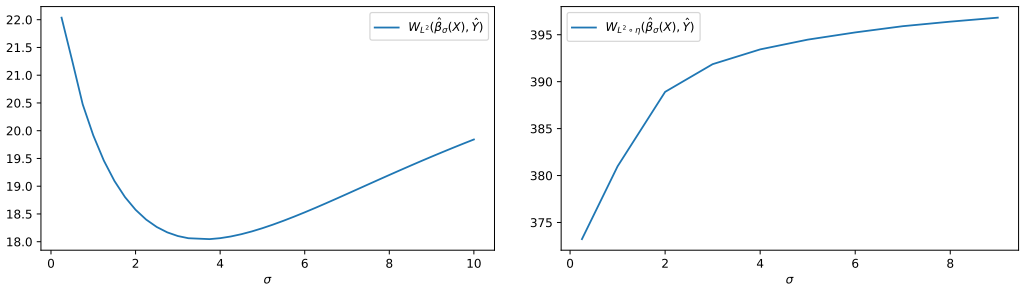

Figure 2: Effect of σ on $W_{d_\mathcal{X}}(\hat{\beta}_\sigma(X), \hat{Y})$ with X and Y obtained from CIFAR-10, for $d_\mathcal{X} = L^2$ (left) and $d_\mathcal{X} = L^2 \circ \eta$ (right)

standard deviation σ, obtaining a blurred dataset $\beta_\sigma(X)$ and corresponding empirical distribution $\hat{\beta}_\sigma(X)$. Figure 2 shows the value of $W_{L^2}(\hat{\beta}_\sigma(X), \hat{Y})$ for varying σ, with similar results observed for other values of p. In particular, we found $W_{L^p}(\hat{X}, \hat{Y}) > W_{L^p}(\hat{\beta}_\sigma(X), \hat{Y})$ whenever $\sigma > 0$. That is, blurring X by any amount brings $\hat{X}$ closer to $\hat{Y}$ in W_{L^p} than not.

To remedy these issues, we sought to replace L^2 with a choice of $d_\mathcal{X}$ that is more naturally suited to the space of images in question. To this end we tried mapping $\mathcal{X}$ through a fixed pre-trained neural network η into a feature space $\mathcal{Y}$, and then computing distances using some metric $d_\mathcal{Y}$ on $\mathcal{Y}$, rather than in $\mathcal{X}$ directly. It is easily shown that, if η is injective, then this defines a metric on $\mathcal{X}$ that we denote $d_\mathcal{Y} \circ \eta$. Moreover, if η is $(d_\mathcal{X}, d_\mathcal{Y})$-continuous, then $d_\mathcal{Y} \circ \eta$ is compact when $(\mathcal{X}, d_\mathcal{X})$ is. We chose η by extracting features from a DenseNet-121 [Huang et al., 2016] pre-trained on ImageNet [Deng et al., 2009], which satisfies both these conditions, so that $W_{d_\mathcal{Y} \circ \eta}$ is a valid metric on $\mathcal{P}(\mathcal{X})$.

To test the performance of $W_{d_\mathcal{Y} \circ \eta}$, we repeated the blurring experiment described above. This time we obtained the plot of $W_{L^2 \circ \eta}(\beta_\sigma(\hat{X}), \hat{Y})$ shown in Figure 2. Happily, we now see that this curve increases monotonically as σ grows in accordance with the declining visual quality of $\beta_\sigma(X)$.

Next, we computed $W_{L^2 \circ \eta}(\hat{A}, \hat{Y})$ over the course of DCGAN training, obtaining the curve shown in Figure 1. In this case we see that $W_{L^2 \circ \eta}(\hat{A}, \hat{Y})$ decreases monotonically towards an asymptote in a way that accurately summarises the visual quality of the samples throughout the training run. Moreover, there is always a large gap between the eventual value of $W_{L^2 \circ \eta}(\hat{A}, \hat{Y})$ and $W_{L^2 \circ \eta}(\hat{X}, \hat{Y})$, which reflects the fact that the GAN samples are still visually distinguishable from real π samples.

We have also observed similar results for an Improved Wasserstein GAN (I-WGAN) [Gulrajani et al., 2017] trained on MNIST and CIFAR-10. We believe this suggests that $W_{L^2 \circ \eta}$ is an appealing choice of D, both due to its nice theoretical properties – metricising weak convergence, and requiring no density assumptions about π – and due to its reliably evaluable empirical performance. We suggest that it would be interesting to use $W_{L^2 \circ \eta}$ to produce a systematic and objective comparison of the performance of all current major GAN implementations in light of (1), and indeed as a metric for guiding future GAN design.

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Sanjeev Arora and Yi Zhang. Do gans actually learn the distribution? an empirical study. *arXiv preprint arXiv:1706.08224*, 2017.

Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and Equilibrium in Generative Adversarial Nets (GANs). *arXiv preprint arXiv:1703.00573*, 2017.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

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.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, pages 271–279, 2016.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

Cédric Villani. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.