Generative Adversarial Networks in 2017

Ben Carr

May 29, 2017

Abstract

Yann LeCunn called them '..the most interesting idea in the last 10 years in ML..' [quo]. Generative adversarial networks (GANs) constitute a powerful new paradigm within unsupervised learning which has garnered a lot of attention since their discovery by Goodfellow et al. in 2014 [GPAM$^+$]. This literature review examines the seminal paper, in particular, the challenges that the paper posed to develop GANs further. An overview of some follow up papers which attempt to extend GANs, theoretically and practically, is given, where it is found that many of the initial challenges have successfully been addressed, especially those involving stability in training.

1 Introduction

Deep learning has made major breakthroughs in unsupervised learning over the last decade, providing deeper, richer data representations. Deep generative models (DGMs) have formed a significant portion of this progress. DGMs aim to learn deep representations from which they can generate new data which approximates the training data. This data can often be complex and high dimensional, like coloured images or video. However, many of these breakthroughs have been stymied by intractability issues. This is commonly because they rely on approximating complex distributions which become intractable at high dimensions and require Markov chain inference [MO].

Generative adversarial networks (GANs), introduced by Goodfellow et al. in 2014 [GPAM$^+$], are a recent form of generative model which sidesteps this difficulty. During training, gradients are solved using exact backpropogation, no inference is required. GANs essentially work by pitting two neural networks against each other, one net, G, generates 'fake' data similar to the training data, and another net, D, tries to discern whether data is fake or not. The pair is analagous to an expert art forger and police detective pitted in opposition. Training these adversary models simultaneously can yield a generative model which generates data with impressive likeness to the training data. In the process of training, latent representations of the dataset can be discovered. To clarify the nomenclature, the *generative* aspect of a GAN comes from the fact that it generates new unseen data and the *adversarial* aspect comes form the fact that two networks are trained against each other.

Since Goodfellow et al.'s original paper, a flurry of extensions and applications to GANs have been developed, across many domains. This review is an effort to take stock of some of this progress. In particular, how the initial problems raised have been addressed. The body of this paper takes the following form. First a brief summary of alternative DGMs is given. This serves to show some advantages of GANs and why they are needed. It also highlights some failings of GANs, in comparison with the other prominent DGM, variational autoencoders. After this a review of the seminal paper of Goodfellow's et al. is provided [GPAM$^+$]. In doing this, a more technical understanding of GANs is gained, in particular, of the initial issues they posed. Then a review of Goodfellow et al.'s follow up paper is given, which addresses some of these challenges, in particular, some technical difficulties of training GANs and in quantifying their performance. An overview of some of the surrounding literature is given which introduces promising extensions to GANs that tackle the fundamental issues they have. These extensions include InfoGAN, DCGAN and EBGAN.

Finally some examples of the many powerful applications of GANs is given, including image generation, video generation and 3D-modelling. This paper will not attempt to exhaustively catalogue these developments. It mainly aims to give a technical overview of the progress in developing the underlying models.

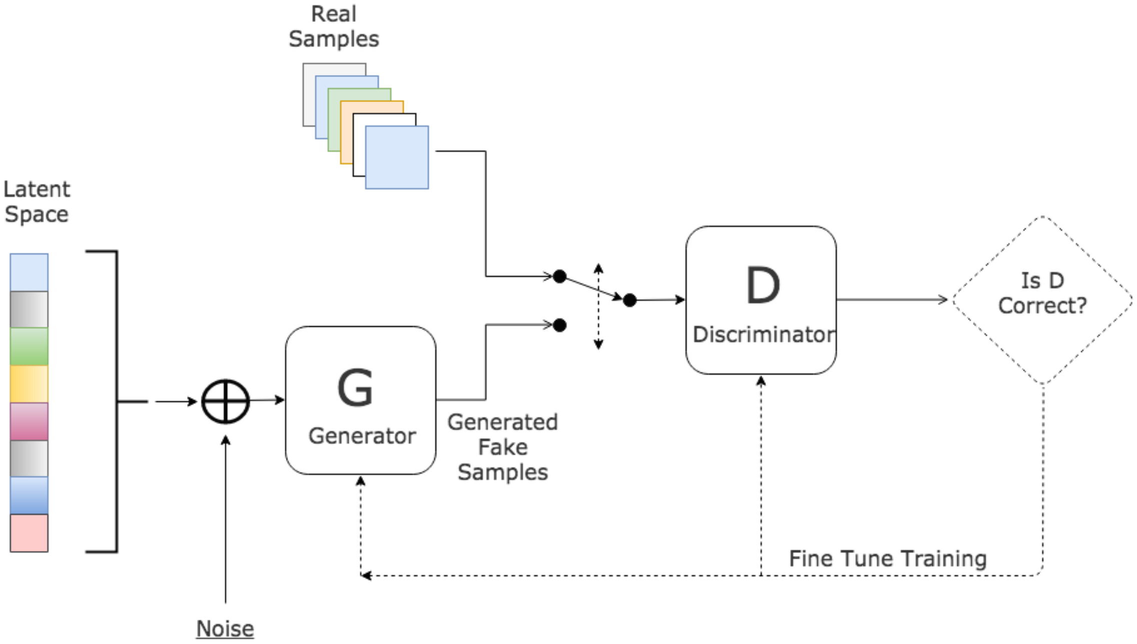

Figure 1: Standard architecture of a GAN [kdn]

To avoid confusion, it is worth noting that *adversarial examples* are a different topic. Adversarial examples are inputs to a model that have been given a very small worst-case perturbation which results in the input being misclassified with high confidence. These examples have been shown to persist across a broad range of models [GSS14]. Adversarial examples do have relevancy with GANs, they highlight a potential limitation of the discriminator net, D, in learning human interpretable features in images [GPAM$^+$]. Further discussion of this is out of the scope of this paper.

2 Standard Generative Adversarial Networks

What follows is an overview of the original GAN architecture proposed by Goodfellow et al. [GPAM$^+$].

To learn the generator's, G's, distribution p_G over the data, an input prior $p_z(z)$ is defined, where z is a noise variable. G is then formally defined as a mapping $G : p_z, \theta_G \to \textit{data space}$ where G is a differentiable multilayer perceptron and θ_G is it's parameters. The second mapping is defined as, $D : \textit{data space}, \theta_D \to (0, 1)$ where D is also a multilayer perceptron, this time with parameters θ_D. Here the output represents the probability that the input was generated by G, i.e., from p_G, as opposed to the original data. A perfect discriminator will have $D(\textit{generated example}) = 0$ and $D(\textit{original example}) = 1$. D is then trained to maximise the probability that it correctly classifies data as either generated or original. Simultaneously, G is trained to minimise $\log(1 - D(G(z))$, and so, minimise the probability that D correctly classifies generated samples. The situation can be described as a *minimax* game with two players and a value function $V(D, G)$.

$$\min_D \max_G V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log(D)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{1}$$

Goodfellow et al. found that early in training, when G is giving poor outputs, the discriminator, D, can too easily identify generated samples and the objective $\log(1 - D(x))$ saturates. Thus it is better to *maximise* $\log(D(x))$ since it has stronger gradients early on, as opposed to *minimising* $\log(1 - D(x))$ [GPAM$^+$]. Another crucial aspect found in training GANs is in allowing G to 'cheat' by looking at the gradients of D and using them in each update step.

Goodfellow et al. go on to prove some basic optimality results about GANs. For example, the first of these results, which on first inspection appears quite enlightening, proves that for fixed G, the optimal discriminator D is

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_G(x)} \tag{2}$$

This is achieved by a simple optimisation in calculus. However this results and the two that follow it are completely irrelevant in practice since they assume that the prior p_G is being optimised, whereas, in practice, it is θ_G that is optimised. Goodfellow et al. do qualify this, claiming that the excellent empirical results '... suggests that they are a reasonable model to use despite their lack of theoretical guarantees'. This might be true, but it nonetheless would be far more desirable to have a strong theoretical foundation. As the theory stands, there is no real insight into why they work as well as they do. This problem is recurrent throughout deep learning and since GANs rely on optimising two deep neural networks by gradient descent, it will likely require a major breakthrough in the theoretical foundations of neural networks in general before a more complete theory of GANs is formed.

Another issue brought up in the paper is the difficulty in training a GAN successfully without it collapsing into a single-point local minima, where G maps too many values of z to the same value of x. This seems to commonly occur when G is trained too often without updating D. This sensitivity to the syncronicity between G and D is commonly highlighted as being a serious problem in training GANs. However, the paper gives no reasons for this sensitivity and no suggestions for guiding the training, either theoretical or heuristic.

3 Improvements in Training

Solving the minimax equation (1) directly is a very difficult task. It is equivalent to finding the Nash equilibrium for a non-cooperative, high dimensional game. Currently, there is no known approach which guarantees to find the Nash equilibrium in such a case [SGZ+]. Thus, GANs are typically trained using gradient descent to find a low value of $V(D, G)$, though, these methods are often not capable of converging. To see how failure to converge might occur, consider a minimax game with $V(G, D) = xy$ where G is trying to minimise $-xy$ while D is trying to minimise xy. It has been shown that gradient descent can enter an orbit, never reaching the optimal $x = y = 0$ [SGZ+].

Motivated by this problem, Salimans et al. propose several techniques with the aim of improving convergence with gradient descent and improving stability, two of these include [SGZ+]:

1. **Feature matching**
   Instead of maximising the output of the discriminator, the objective for the generator is changed to match the statistics of the real data. The discriminator is trained to specify the statistics that are deemed worthy of matching. More precisely, G is trained to match the expected value of the features of an intermediate layer of D. D is trained to find features which are most discriminative of real data against generated data.

   This technique has been shown to improve stability as the generator is prevented from overfitting on the current discriminator [SGZ+].

2. **Minibatch discrimination**
   As mentioned in section 2, GANs are susceptible to collapse into a single-point, local minima, where G continually outputs the same value. This occurs because the gradient of D points in the same direction for many similar inputs. Since the discriminator processes each sample independently, there is no process of making the gradients dissimilar and allowing G to generate more realistic data.

   Minibatch discrimination overcomes this problem by allowing the discriminator to process multiple inputs at once, giving information on how *close* a particular point is to others preceding it. Although, as before, D still outputs a single scalar, signifying the probability of a single sample being generated or not. Only now D can discourage similar data being generated by G [SGZ+].

3.1 Assessment of Image Quality

One difficulty with training GANs is the lack of a clear objective function to optimise. For example, there is not an obvious way of quantifying how realistic an image is. To overcome this problem, Salimans et al. used Amazon's Mechanical Turk service to hire human contractors to discern between real and generated data. A problem with this approach is the variability in the quality of assessments given. This depended on at least three factors; the assessors motivation (e.g., amount of money rewarded), the set-up of the task (e.g., the wording used) and the amount of feedback given to the assessors (the more feedback, the better the ability to discern real images). They found it key to have a high number of samples, suggesting at least $50k$ [SGZ$^+$].

4 Alternative Approaches

Two alternative deep generative models (DGMs) will be outlined here. The first preceeded GANs while the second is their contemporary.

4.1 Deep Boltzmann Machines

Deep Boltzmann machines (DBMs) have probably been the most successful DGM until recently [SH]. The one major problem of DBMs and similar approaches is that their training relies on maximising an intractable negative log liklihood. Such models typically rely on numerous approximations to estimate their liklihood gradients. As noted in Goodfellow et al., this is what spurred on development of DGMs which rely on exact backpropogation, such as GANs or VAEs[GPAM$^+$].

4.2 Variational AutoEncoders

Variational AutoEncoders (VAEs) are another form of DGM which have attracted a lot of attention in recent years. They have a similar architecture to a standard autoencoder network, one crucial difference is a noise term included before the decoder, in latent space. This extra prior adds some control to the latent representation distribution. One major advantage of this setup is the possibility of embedding new data into the latent space of a trained VAE. This allows decoding of new data, opening the door to a range of applications, such as 3D-embedding the expressions of one face onto another [VAE].

A drawback of standard GANs is that they do not allow embedding data into latent space in this way. Another advantage that VAEs have is the clear way of evaluating their performance, by log-liklihood. There is not such a straightforward way to measure the quality of a GAN other than visualising the samples generated. This is largely due to the fact that GANs do not have a straightforward objective function to optimise. Furthermore, GANs are much more difficult to train. Given the potential instabilities that can arise from the minimax game (equation 1), much care must be taken in choosing how to balance the training of the generator against the training of the discriminator [SGZ$^+$].

One major disadvantage of VAEs is that the injected noise gives a noisy reconstruction of the input. For images this often leads to blurring. GANs typically enjoy significantly crisper results if they are trained right. Another advantage of GANs is the relatively simple nature of their latent space. This allows interpolation between latent data points and synthesising of more diverse artificial data. VAEs have difficulty performing simple linear algebra on their latent space since they have a more complex prior.

Three disadvantages of GANs have been highlighted here: inability to embed data into latent space, instability of training and difficulty of evaluating their performance. This paper will discuss the development in addressing these problems.

5 Theoretical Extensions

Here, a non-exhaustive list of extensions to the original GAN architecture will be outlined. Many of these address problems highlighted in Goodfellow et al [GPAM$^+$].

5.1 DCGAN

Radford et al. propose a new architectural topology of Convolutional GANs that makes them more stable during training, they name this class Deep Convolutional GANs (DCGAN) [RMC15]. As well as generating realistic bedroom images, they show that a pre-trained DCGAN performs competitively against other unsupervised models on image classification. Furthermore, they show that both the discriminator and the generator learn useful representations, like beds and windows.

The architectural changes they make to the initial GAN framework are more restraints than new features. They include: replacing pooling layers with convolutional strides, use of batchnorm, removal of fully connected hidden layers and ReLU activation for all but the output which uses Tanh.

5.2 EBGAN

Instead of using the discriminator as a probabilistic binary model, Zhao et al. have shown that GANs can be successfully trained using an energy function for D [ZML]. The discriminator for an energy-based GAN (EBGAN) assigns low energies to regions of high data density and high energies to regions of low data density. The discriminator no longer has an explicit probabilistic interpretation, it can instead be interpreted as a trainable cost function for the generator. Zhao et al. show that this new formulation is competitive with the current best GAN results in generating realistic images after training on MNIST, the CelebA face dataset or the LSUN bedroom dataset.

After training on ImageNet however, the generated samples are unrealistic. For instance, dogs are often generated as shapeless balls of fur. This is likely due to a lack of data for each category and a much wider diversity of categories to represent. One thing the EBGAN did well was to identify certain scene dynamics, like the fact that objects generally appear in the foreground and certain textures are generally background components, like sea or buildings.

5.3 InfoGAN

Another interesting extension is InfoGAN, proposed by Chen et al [CDH$^+$16]. This is an information-theoretic extension. The general idea is to maximise the mutual information (the amount of information learned about one variable by knowing the other) between a fixed small subset of the GANs noise variables (a noise variable, z, is the input of the generator) and it's outputs ($G(z)$). To enforce this new constraint, a regularisation term is introduced into the minimax game (original equation 1). Thus, the training process doesn't just encourage generation of realistic samples but also of a high amount of mutual information between noise variables and outputs for the generator.

This extra objective forces the network to discover salient relationships between the noise and the output. For example, consider the task of generating handwritten digits after training on the MNIST dataset. InfoGAN can interpret latent codes in the noise variables which correspond to the 10 classes, 0-9. It can also identify latent codes for thickness of stroke and angle of rotation. Remarkably, this entirely unsupervised approach is competitive with state-of-the-art supervised approaches in learning salient representations of a dataset. This is hugely beneficial in disentangling meaning from the vast amounts of unlabelled data that is available.

6 Practical Applications

Despite the recency of GANs, there have been a variety of successful applications. Here, a cherry picked sample will be outlined. Thus far, the most common application has been in image synthesis.

6.1 Image Generation

1. **Text to Image**
   There have been several papers presenting realistic text to image generation with the use of GANs. For example, Reed et al. successfully trained a GAN to generate realistic bird and flower images from unseen text descriptions [RAY$^+$16]. This result and others like it have issues generating images of high resolution. In this regard, the most impressive result has come from a stacked GAN (StackGAN) presented by Zhang et al. [ZXL$^+$]. Building on the success of Denton et al.'s Laplacian Pyramid framework (LAPGAN), they implemented a 2-stage process to overcome the resolution issue. First, images of the rough shapes and basic

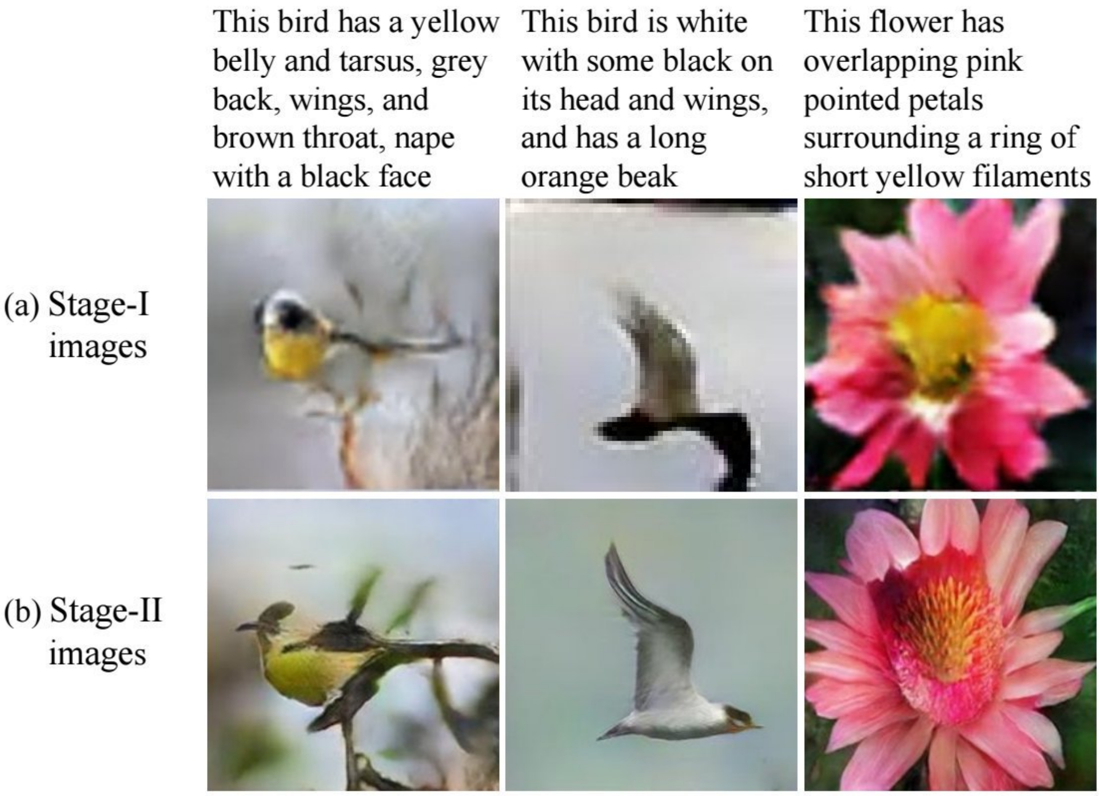
This bird has a yellow belly and tarsus, grey back, wings, and brown throat, nape with a black face
This bird is white with some black on its head and wings, and has a long orange beak
This flower has overlapping pink pointed petals surrounding a ring of short yellow filaments
(a) Stage-I images
(b) Stage-II images

Figure 2: The 2-stage StackGAN process [ZXL+].

colours of objects are generated at low resolution. These are then passed through a second GAN which yields photo realistic images at a much higher resolution (256 × 256 as compared to LAPGAN's 96 × 96).

2. **Texture Synthesis**
   The task of layering a texture over an image in an aesthetically pleasing way, i.e., stylizing it, is very challenging. GANs, or an architecture very similar, have recently made impressive results. Ulyanov et al. presented a GAN-esque model capable of this, the main modification being the replacement of a discriminator with a *descriptor* network. The descriptor is not trained but instead calculates hand-picked statistics of the stylised image which the generator tries to match during training [ULVL]. The results of this architecture are state of the art on most styles and images although it has difficulties in certain areas, for example, the panda in figure 3 can be synthesised better using competing techniques.

3. **Super Resolution**
   Ledig et al. presented SRGAN (super resolution GAN) which can achieve photo realistic 4× upscaling, i.e., it can effectively increase the resolution of an image by a factor of 4 [LTH+16]. The discriminator was trained to discriminate between super-resolved images produced by the generator and original photo-realistic images.

6.2 Video Generation

GANs have proven to achieve state of the art results in video synthesis too. Video synthesis, where inputs are preceding frames and outputs are succeeding frames, is a much more challenging task than image generation. For a model to do it well, a spatial awareness of objects, their actions and their 3D nature is often necessary. Because of this inherent difficulty, the sub-field is still in it's infancy. Development here is particularly desirable since an A.I. that can generate realistic video of the future, can effectively predict future states - a central pillar of intelligence. GANs offer a powerful unsupervised tool in tapping into hidden representations latent in large video datasets.

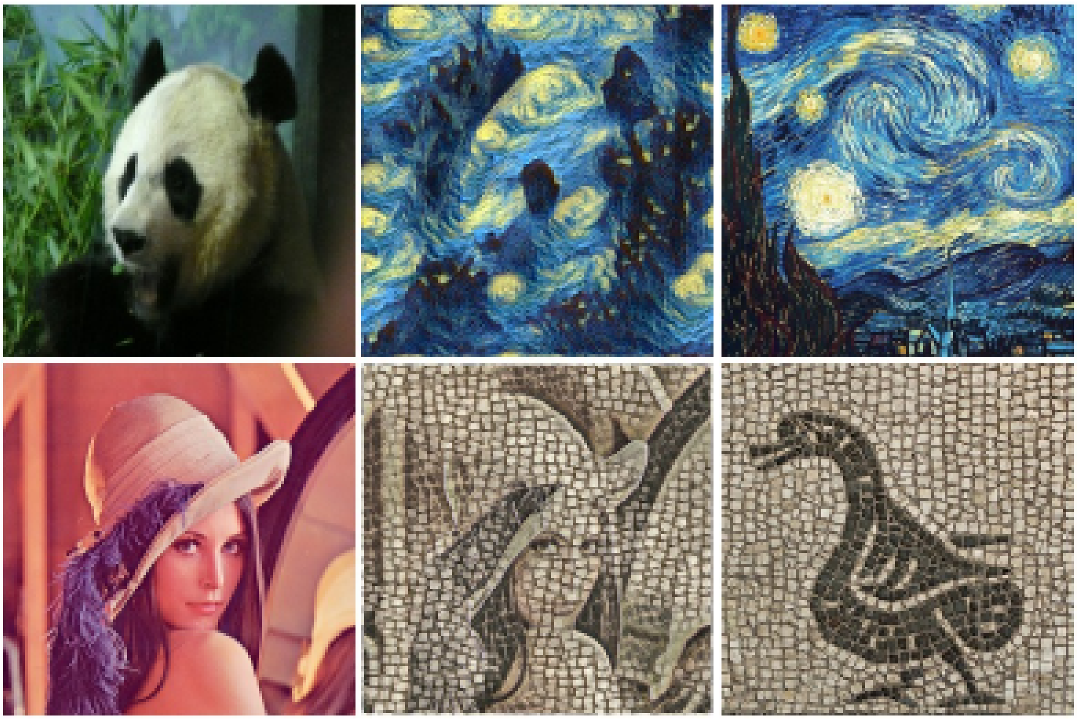

Figure 3: Texture synthesis using GAN-esque architecture. Original image (left), synthesised image (middle), original style (right) [ULVL].

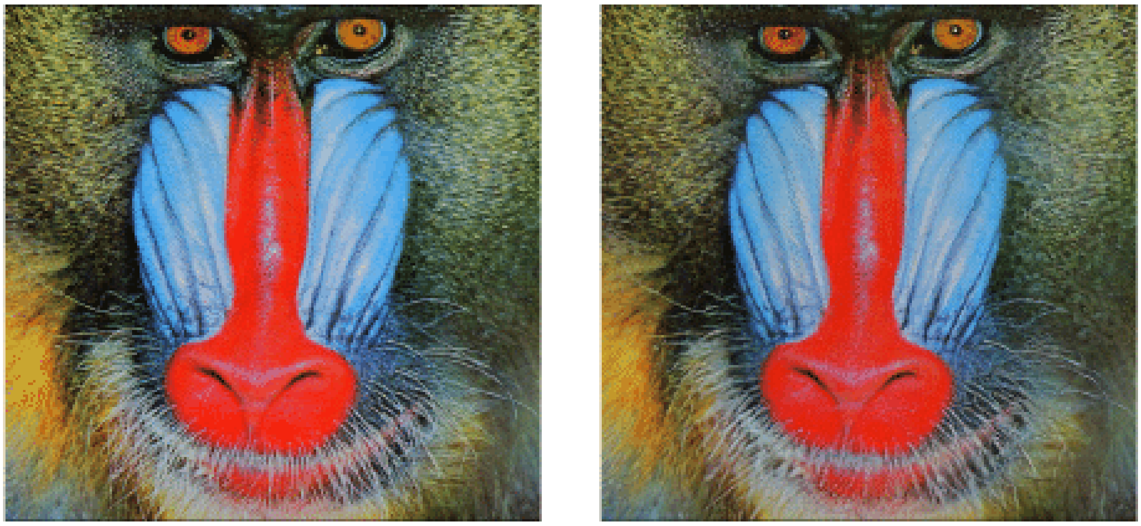

Figure 4: SRGAN generated image (left) is nearly identical to upscaled original (right) [LTH+16].

Mathieu et al. have shown that GANs significantly outperform MSE (mean square error) methods in predicting one or two frames into the future, yielding much sharper results [MCL]. Ranzato et al. have shown that GANs can also be employed to fill in missing frames [RSB+14]. Perhaps the most promising development has been with the use of scene dynamics, as shown by Vondrick et al [VPT]. By training a two-stream GAN, which generates the foreground and background separately, the network is forced to identify dynamic foreground objects in contrast with the static background. By employing this strategy, up to 32 future frames can be extrapolated. Compared to the previously mentioned strategies this a marked improvement.

This progress is substantial but there are still large issues. The number of frames of extrapolation is still too small and often unrealistic events can occur in those frames that are generated [VPT]. More powerful generative models will be required, whether this will come in the form of adversarial networks is to early to say.

6.3 Other Applications

Interesting applications of GANs in 3D modelling have been presented [WZX+] [Cha], also, in semi-supervised learning (training with partially labelled data) [SGZ+] [RMC15] [ZML].

7 Conclusions and Future Work

Since the seminal paper by Goodfellow et al. many of the initial problems with GANs have been addressed [GPAM+]. The main issue brought up then was the difficulty in training GANs; failure of convergence, single point collapse, sensitivity to synchronisation of training speeds between G and D. These issues have been tackled by a variety of techniques. Some minor tweaks like feature matching or minibatch discrimination have worked well in preventing single point collapse and in synchronisation of training G and D [SGZ+]. Perhaps more pertinent are the major theoretical extensions to the original GAN formulation. EBGAN provides a different, insightful way of interpreting the discriminator, as an energy based cost function as opposed to a probabilistic binary model. InfoGAN offers a powerful, information-theoretic way of finding latent classifications or attributes in a dataset, like the digit number or stroke thickness in MNIST.

These developments, within the 18 months since their discovery, GANs have come to represent a large step forward in unsupervised learning. On virtually all benchmarks GANs obtain state of the art results (with generally VAEs being the only other contender). For most datasets, there is inherent difficulty in generating unseen, realistic samples by unsupervised learning, for some cases the task is AI-complete (a solution would take nothing less than a strong AI). No generative model is close to achieving strong-AI levels of intelligence. Thus, despite the success GANs have had, there is still a very long way to go. To advance, much more powerful models must be developed to better capture latent representations.

A stronger theoretical foundation of GANs and of deep learning in general would help create stronger models. For instance, if there was a proven optimal pace for training G against D, if there was an analytically closed solution to solving the Nash Equilibrium or if there was a better understanding of how gradient descent worked in training deep nets, it would benefit 'up-stream' GAN development greatly.

Another potential way of creating more powerful GANs would be to combine the different successful extensions. There is currently no work combining InfoGAN, EBGAN and StackGAN for instance. As seen in many other areas of machine learning, combining techniques like this can be of huge benefit.

References

[CDH+16] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. jun 2016.

[Cha] Learning to generate chairs with CNNs.

[GPAM+] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets.

[GSS14] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. dec 2014.

[kdn] Generative Adversarial Networks – Hot Topic in Machine Learning. http://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html. Accessed: 2016-12-12.

[LTH$^+$16] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. sep 2016.

[MCL] Michael Mathieu, Camille Couprie, and Yann Lecun. DEEP MULTI-SCALE VIDEO PREDICTION BEYOND MEAN SQUARE ERROR.

[MO] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets.

[quo] What are some recent and potentially upcoming breakthroughs in deep learning? https://www.quora.com/What-are-some-recent-and-potentially-upcoming-breakthroughs-in-deep-learning. Accessed: 2016-12-12.

[RAY$^+$16] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative Adversarial Text to Image Synthesis. may 2016.

[RMC15] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. nov 2015.

[RSB$^+$14] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. dec 2014.

[SGZ$^+$] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved Techniques for Training GANs.

[SH] Ruslan Salakhutdinov and Geoffrey Hinton. Deep Boltzmann Machines.

[ULVL] Dmitry Ulyanov DMITRYULYANOV, Vadim Lebedev VADIMLEBEDEV, Andrea Vedaldi, and Victor Lempitsky LEMPITSKY. Texture Networks: Feed-forward Synthesis of Textures and Stylized Images.

[VAE] Variational Autoencoder in Tensorflow – facial expression low dimensional embedding. http://int8.io/variational-autoencoder-in-tensorflow/. Accessed: 2016-12-12.

[VPT] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating Videos with Scene Dynamics.

[WZX$^+$] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling.

[ZML] Junbo Zhao, Michael Mathieu, and Yann Lecun. ENERGY-BASED GENERATIVE ADVERSARIAL NET- WORKS.

[ZXL$^+$] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris Metaxas. StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks.