A Review of Recent Improvements of Generative Adversarial Networks

Benjamin Wilhelm[0000−0003−1608−5267]

University of Konstanz, Germany
benjamin.wilhelm@uni-konstanz.de

Abstract. GANs are pairs of neural networks that train by competing against each other. They got a lot of attention in recent years—especially in image generation. In this paper, I will first explain the general idea of GANs. Later I will review approaches to solve their most common issues like generating high-resolution images, the variation of generated images, and assessing of results. I will focus on approaches introduced by the Progressive Growing of GANs paper by Karras et al. [9].

Keywords: Deep learning · Generative adversarial nets · Progressive growing

1 Introduction

Generative Adversarial Networks [4] are a great fit to generate images. They have been used in a wide variety of image analysis tasks like generating images from given text snippets or attribute descriptions [17,21], superresolution [11,20] and image inpainting [15,23,7,22].

With the general formulation of Generative Adversarial Networks, there are some problems that arise like generating images with a high-resolution, variation of the generated images and assessing the results. After introducing the general formulation of GANs I will review approaches that have been introduced to solve these issues while focusing on Progressive Growing of GANs [9]. I will keep the mathematical details to a minimum and focus on intuitive explanations.

2 Generative Adversarial Nets

Goodfellow et al. introduced a general framework for training generative models [4] that make use of a generator model G and a discriminator model D. The goal of the generator model is to produce samples that are indistinguishable from the samples of the training distribution. The goal of the discriminator model is to distinguish samples produced by the generator model from samples from the training distribution. This leads to a game between generator and discriminator that can be easily explained by the following analogy by Goodfellow.

The generator model can be seen as a counterfeiter and the discriminator as the police. The counterfeiter is trying to counterfeit money that can not be

detected by the police. The police are trying to get better in detecting the fake money. If the police get better in detecting fake money the counterfeiter has to get better in faking it. If the counterfeiter gets better in faking money the police have to get better in detecting it. This game eventually leads to a state where the fake money by the counterfeiter is indistinguishable from real money.

2.1 Training Procedure

The training procedure consists of two parts that are repeated. During the first part, only samples of the real data distribution and generated samples are presented to the discriminator and the discriminator is updated by ascending its stochastic gradient. This is repeated k times. During the second part, the generator is updated by presenting samples of it to the generator and computing its gradient.

3 Problems and Solutions

In this section, I describe problems that exist with training GANs like it was introduced by Goodfellow and solutions that were proposed in the literature. I will focus on solutions introduced by Karras et al. [9]. To most of the problems, there is a wide variety of possible approaches that I won't cover completely.

3.1 High-Resolution Images

An important problem when generating images using GANs is that GANs struggle with high-resolution images [14].

This problem can be addressed by an architecture and training procedure that progressively increases the resolution of the generator and the discriminator model. The generator model consists of multiple blocks that increase the resolution. The fist bock takes the latent random vector as input and outputs a feature map with a spacial resolution of 4×4. The next bock operates on an upsampled version of this feature map with twice the spatial resolution. Blocks like these are stacked on each other until the desired resolution is reached. The discriminator model is basically a mirror image of the generator model and the feature maps are downsampled between the blocks.

This kind of architecture was already introduced by Chen et al. in 2017 [14]. Karras et al. [9] introduced a new training procedure that utilizes that lower resolution GANs are easier to train. The new training procedure reduces the training time while improving the quality of the output image.

In the first part of the training the first block of the generator (which outputs 4×4 feature maps) is connected to a 1×1 convolutional layer (that transforms the features into RGB values) and the last block of the discriminator gets its input from a similar layer (that transforms RGB values into features). These networks form a GAN on its own and are trained with respect to downsampled images from the training distribution.

In each following training part, the following block of the generator is slowly faded into the network while the weights of the previously trained blocks are copied but are still trainable. The block is faded in by adding its RGB output p_i to an upsampled version of the output of the previous block p_{i-1} weighted with a parameter α $(\alpha \cdot p_i + (1-\alpha) \cdot p_{i-1})$ which is increased linearly. Analogously, the previous block of the discriminator is added and slowly faded in and the whole model is trained with respect to images from the training distribution downsampled to the current resolution. This training procedure enables the training of high-resolution GANs that will be demonstrated in Section 4.

3.2 Variation of Generated Images

Another problem of GANs is that they tend to generate images with lacking variation.

A rather easy method to force the generator to generate a larger variety of images is to give the discriminator information about the variety of the current minibatch. Note that the minibatch can be generated by the generator or drawn from the training discriminator. Because the discriminator can use this information to make its prediction the generator is forced to generate minibatches that have a similar variety to the minibatches of the training distribution.

Karras et al. [9] compute the standard deviation of each feature in each spatial location over the minibatch and average the values to arrive at the average standard deviation over the minibatch. This value is concatenated to the input of the last block of the discriminator network as an additional constant feature map which allows the discriminator to make use of the value.

This is a simplified version of "minibatch discrimination" [18] which computes multiple feature statistics of the minibatch. It is neat because it introduces no additional parameters (except a few more parameters that require no special handling in the convolutional layer after the feature map is appended) and no additional hyperparameters.

Other solutions have been proposed to increase the variation of the generated examples. *MAD-GAN* [3] uses multiple generators and the discriminator must also identify the generator which forces the generators to produce diverse images. Unrolled GANs [13] unroll the discriminator which (they claim) also increases the diversity.

3.3 Assessing Results

Comparing GANs to each other is hard to do and usually requires manual comparison which can be difficult and subjective. Therefore automated methods that compute a metric on how well the generated examples fit the training distribution is desirable. This metric can then be compared between different GANs.

Karras et al. [9] introduce an evaluation metric that computes the sliced Wasserstein distance [16] between image patches sampled from different levels of a Laplacian pyramid (each level of the pyramid contains the structure of

one specific frequency band). A small Wasserstein distance indicates that the training patches and generated patches appear similar. This means that a small Wasserstein distance for higher resolutions of the pyramid means that high-level features like edges and noise are similar. A small Wasserstein distance for lower resolutions of the pyramid indicate that the overall structures are similar.

Another metric is the Inception Score (IS) [18] which evaluates the generated images based on how well a the pretrained Inception model [19] can predict a class label and how much the classes vary. Good generated images have a low entropy on their prediction (The model can predict on class very clearly) and a high entropy on the marginal distribution (All classes get predicted equally often). The Inception Score should only be used on models trained on ImageNet and there are more issues pointed out by Barratt and Sharma [2].

The Fréchet Inception Distance (FID) [6] is an improved version of the IS that uses the features computed by an intermediate layer of the Inception model and a multivariate Gaussian distribution. Therefore, it is more robust against cases where the model only predicts one image per class.

Despite all the metrics that have been introduced the evaluation of GANs is still a big issue because none of the introduced metrics is a clear fit for all cases.

3.4 Other Problems and Improvements

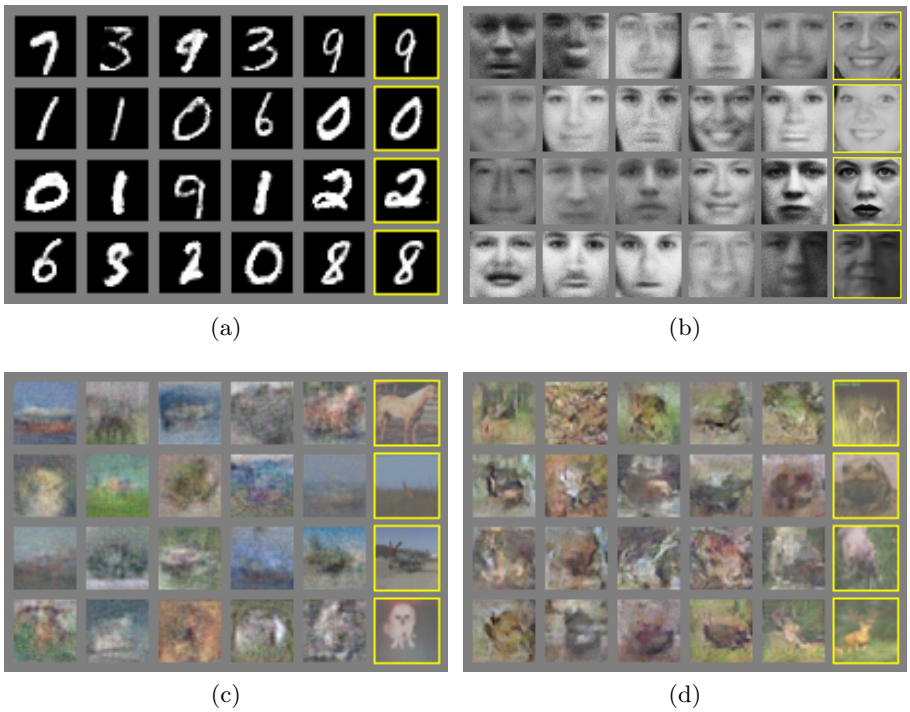
In the early stages of the training, the generated images and the images from the training distribution can be to easy to distinguish which can lead to gradients for the generator that point in random directions. This can be improved by using a more stable loss function than the Jensen-Shannon divergence (that was used in the original formulation of GANs [4]) like least-squares [12] and Wasserstein distance [1,5].

Another problem that appears if the competition between the generator and the discriminator model is not fair (e.g. the discriminator is trained is much better than the generator) are escalating signal magnitudes. This issue can be counteracted with Batch Normalization [8] but Karras et al. introduced another method where they just normalized each feature vector after each convolutional layer to unit length [9].

4 Evaluation

In this section, I will show some results from the original GAN paper [4], Progressive Growing of GANs [9] and Style-Based GANs [10].

Figure 1 shows results by Goodfellow et al. [4]. We can see that only MNIST images could be generated such that they look like real MNIST images. The examples generated for the Toronto Faces Dataset are still recognizable as faces but most of them have obvious artifacts. The images generated for the CIFAR-10 dataset are not really recognizable but are visually similar to images from the dataset.



(a) (b) (c) (d)

Fig. 1. Results from Goodfellow et al. [4]. The rightmost column shows the closest training example to the generated example beside it. The used datasets are (a) MNIST (b) TFD (c) CIFAR-10 (fully connected model) (d) CIFAR-10 (convolutional discriminator and deconvolutional generator)

Note that all generated images by the original GANs have a very low resolution. Judging from the results from the CIFAR-10 dataset the method wouldn't work for images with higher resolutions without addressing this issue in network architecture and training procedure like described in Subsection 3.1.

Figure 2 shows results by Karras et al. [9] on the high-resolution CELEBA-HQ dataset. The generated images have a resolution of 1024×1024 and we can observe that the model is able to generate fine details like hair and beards in fitting locations. Also the general structure of the images is realistic—one could be fooled into thinking that the images are photographs.

To verify the quality of the generated images I downloaded the released source code and model weights and generated random samples myself which can be found in Figure 3. We can observe that the samples generated from the CELEBA-HQ dataset (Figure 3a) look significantly worse than the carefully picked samples presented in the paper but still include fine details and mostly overall realistic structure. The model seems to have the biggest issues with outlines and background. This shows that the model can't yet reliably generate

Fig. 2. Results from Karras et al. [9] from the CELEBA-HQ dataset.

images that can fool humans and that the selection of images in the paper does not represent a random selection. Images of lower quality can only be found in the appendix of the paper. Additionally, I generated images from the using the provided model weight for the cat category of the LSUN dataset which can be found in Figure 3b. We can see that the overall structure of these images does not look realistic. There are parts in the images that look like parts from cats but the parts are not composed correctly. This could be due to a higher variety of training examples with respect to there low-frequency structures. Therefore the model might not learn a set of realistic structures during the first parts of the training on low-resolution images.

Lastly, I show generated images by Karras et al. [10] using a style based generator architecture in Figure 4. The generated images for the FFHQ dataset look even more promising than the image generated by progressive GANs on the CELEBA-HQ dataset and the cats also look better. Since there is no source code and weights available for style based GANs I can not compare the images with a random set.

5 Conclusion

I explained the general idea of GANs and presented some of the most important fundamental issues. I presented the approaches to solving these issues introduced in the Progressive Growing of GANs paper and gave a few impressions on other approaches. I showed results from Goodfellow and Karras and evaluated them critically with respect to the visually recognizable problems. Additionally, I generated own random examples using the source code and network parameters from the Progressive Growing of GANs paper and realized that the images pre-

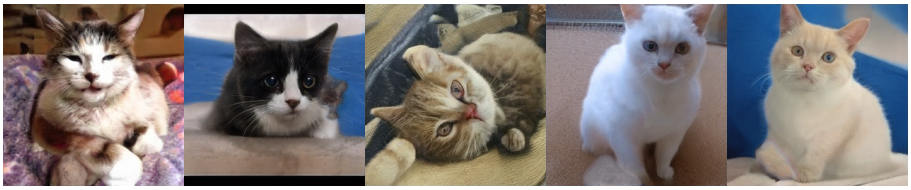
Fig. 3. Generated images using the released source code and model from Karras et al. [9] from (a) the CELEBA-HQ dataset (b) LSUN cat category.

sented in the main part of the paper give a false impression of the quality of the model but that it is still very powerful.

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017)
2. Barratt, S., Sharma, R.: A note on the inception score. arXiv preprint arXiv:1801.01973 (2018)
3. Ghosh, A., Kulharia, V., Namboodiri, V.P., Torr, P.H., Dokania, P.K.: Multi-agent diverse generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8513–8521 (2018)
4. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)

(a)



(b)

Fig. 4. Results from Karras et al. [10] from the (a) FFHQ dataset and (b) LSUN cat category.

5. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 5767–5777. Curran Associates, Inc. (2017), http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf
6. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems. pp. 6626–6637 (2017)
7. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. ACM Transactions on Graphics (TOG) **36**(4), 107 (2017)
8. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
9. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=Hk99zCeAb
10. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. arXiv preprint arXiv:1812.04948 (2018)
11. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. arXiv preprint (2017)

12. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2794–2802 (2017)
13. Metz, L., Poole, B., Pfau, D., Sohl-Dickstein, J.: Unrolled generative adversarial networks. arXiv preprint arXiv:1611.02163 (2016)
14. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 2642–2651. JMLR. org (2017)
15. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2536–2544 (2016)
16. Rabin, J., Peyré, G., Delon, J., Bernot, M.: Wasserstein barycenter and its application to texture mixing. In: International Conference on Scale Space and Variational Methods in Computer Vision. pp. 435–446. Springer (2011)
17. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. In: Balcan, M.F., Weinberger, K.Q. (eds.) Proceedings of The 33rd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 48, pp. 1060–1069. PMLR, New York, New York, USA (20–22 Jun 2016), http://proceedings.mlr.press/v48/reed16.html
18. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in Neural Information Processing Systems. pp. 2234–2242 (2016)
19. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
20. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Loy, C.C.: Esrgan: Enhanced super-resolution generative adversarial networks. In: European Conference on Computer Vision. pp. 63–79. Springer (2018)
21. Yan, X., Yang, J., Sohn, K., Lee, H.: Attribute2image: Conditional image generation from visual attributes. In: European Conference on Computer Vision. pp. 776–791. Springer (2016)
22. Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., Li, H.: High-resolution image inpainting using multi-scale neural patch synthesis. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 1, p. 3 (2017)
23. Yeh, R., Chen, C., Lim, T.Y., Hasegawa-Johnson, M., Do, M.N.: Semantic image inpainting with perceptual and contextual losses. arxiv preprint. arXiv preprint arXiv:1607.07539 **2** (2016)