REPRODUCING PA-GAN

Daniel Fojo*, Paula Gómez*, Janna Escur* & Miquel Tubau*

Image Processing Group (GPI)
Universitat Politecnica de Catalunya (UPC)
Barcelona, Spain
daniel.fojo@estudiant.upc.edu,
{paula.maria.gomez,janna.escur,miquel.tubau}@alu-etsetb.upc.edu

Xavier Giró
Image Processing Group (GPI)
Universitat Politecnica de Catalunya (UPC)
Barcelona, Spain
xavier.giro@upc.edu

ABSTRACT

One of the challenges in machine learning research is to ensure that published results are reliable and reproducible. In support of this, the ICLR Reproducibility Challenge[1]has been set up in order to investigate the reliability of empirical results submitted to the 2019 International Conference on Learning Representations (ICLR). This work aims to reproduce the results of the paper "PA-GAN" which can be found in openreview[2].

The original paper (PA-GAN) proposes a novel method (Progressive Augmentation - PA) in order to improve the performance of a Generative Adversarial Network (GAN) by hindering the task of the Discriminator to avoid its rapid convergence. In this report, we show what we understood from the original paper how did we implement the main functionalities. Besides, we also demonstrate the reliability of their experiments by showing either the achieved results or the small issues that were found, which later were corrected by contacting the authors. Even that, some of the results were not completely reproducible. However, in the last section of this report we also state the main drawbacks found and how did we solve them.

Our code is openly available here https://github.com/telecombcn-dl/2018-dlai-team1.

1 INTRODUCTION

Being able to reproduce the results of a paper is one of the main challenges in machine learning research. This is due to the reason that it is really important to ensure that published results are reliable and reproducible. In support of this issue, the ICLR Reproducibility Challenge has been set up in order to investigate the reliability of empirical results submitted to the 2019 International Conference on Learning Representations (ICLR).

For this challenge we chose to reproduce PA-GAN, a paper submitted to ICLR 2019 which we explain in the following section 2. In section 3, we explain in detail our implementation of Progressive Augmentation. After that, in section 4, the results achieved are shown.

*Equal contribution
[1]https://reproducibility-challenge.github.io/iclr_2019/
[2]https://openreview.net/pdf?id=ByeNFoRcK7

2 PA-GAN

The original paper introduces a new technique to improve Generative Adversarial Networks (GANs) performance during training. The paper states that despite recent progress, GANs still suffer from training instability thus requiring careful consideration of the architecture design and hyper-parameter tuning. It is known that the reason for this fragile training behaviour is, partially, due to the discriminator performing very well very quickly. As a consequence, its loss converges to zero thus providing no reliable backpropagation signal to the generator (Zhang & Khoreva, 2019). So, taking this issue into consideration, the authors of the original paper introduce Progressive Augmentation, described as a novel approach to increase the discriminator's stability during training. The main idea behind this technique is to augment the difficulty of the discriminator's task in order to balance the adversarial training. In order to make it possible, PA-GAN structurally augments both fake and real training samples and, then, it minimizes the divergence between the distributions defined in the augmented sample space. The minimized divergence is computed by using the adversary process introduced in Goodfellow et al. (2014).

2.1 Single augmentation level

In a typical GAN, the task of the discriminator consists on classifying real data samples (x_d) into the TRUE class and, generated samples (x_g), produced by the generator, into the FAKE class. When introducing PA to these samples, they are augmented by a bit $s \in \{0, 1\}$ and a new classification can be done. The new $(x_d, s = 0)$ and $(x_g, s = 1)$ will belong to the TRUE class while $(x_d, s = 1)$ and $(x_g, s = 0)$ will belong to the FAKE one. Thus, as it can be seen in figure 1, real samples will not belong just to the TRUE class and, analogously, generated samples will not belong just to the FAKE one.

To determine the true value of a sample, we will consider the real and synthetic samples to convey one bit of information, x_d encoding a 0 and x_g encoding a 1. Thus, the checksum of a pair (x, s) determines the respective class, i.e. checksum zero for TRUE and one for FAKE. This new checksum, poses a more difficult task for the discriminator, since it has to classify both TRUE and FAKE samples while also computing the checksum operation. In (Zhang & Khoreva, 2019) it is stated that this prevents early maxing-out of the discriminator without compromising the task of the generator.

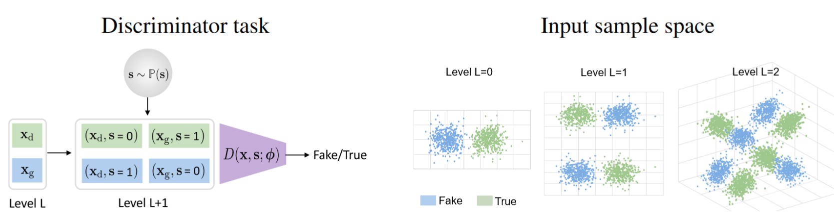


Figure 1: Visualization of progressive augmentation. With each extra augmentation level ($L \rightarrow L + 1$) the dimensional of the discriminator input space is increased and the discrimination task gradually becomes harder. This strategy prevents the discriminator from easily finding a decision boundary between two classes and thus leads to meaningful gradients for the generator updates. Figure taken from (Zhang & Khoreva, 2019)

2.2 Progressive multi-level augmentation

Single level augmentation is extended by changing s to be an arbitrarily long sequence of bits $\boldsymbol{s}$. Then, the augmented discriminator takes the bit sequence $\boldsymbol{s}$ as well as the sample x. Following the procedure from the single level case, the same checksum mechanism remains. Namely, the discriminator has to retrieve one bit of information carried by x, and perform the checksum operation with the arbitrarily long random sequence $\boldsymbol{s}$. The original paper claims that the difficulty of this task grows as the length of $\boldsymbol{s}$ grows. This arises the idea of increasing the length of $\boldsymbol{s}$ every time that the discriminator becomes too powerful. Moreover, it is the consistency of the checksum mechanism across different augmentation levels that allows progressive augmentation.

The same discriminator can be trained from a lower augmentation level and gradually take more bits into consideration (see Figure 2).

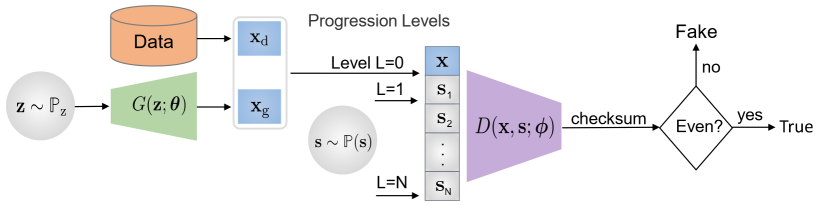


Figure 2: Overview of PA-GAN, and how the discriminator's task difficulty grows with the length of $\boldsymbol{s}$. Figure taken from (Zhang & Khoreva, 2019)

3 IMPLEMENTATION

For our implementation, we use as a baseline a Spectral Normalized-DCGAN (Miyato et al., 2018), as it is used in the original paper. Then, in order to add Progressive Augmentation to the network, the generator can remain unchanged while the discriminator has to accommodate the increase of the samples with $\boldsymbol{s}$. To this end, we modify the first layer of the discriminator every time a level is increased. The only modification are the number of input channels. The kernel size, the padding and the stride of this layer, are all maintained.

The bit sequence $\boldsymbol{s}$ is pre-processed into a form compatible with x. For the case of images as an input, each bit of the sequence is translated to an image channel full of 1s or 0s.

We use non-saturating loss (Goodfellow et al., 2014) as they do in the original paper. However, when they explain the progression scheduling used to increase the augmentation level, we think that it is not clearly expressed. When they state: "If the current KID score is less than 5% of the average of the two previous ones attained at the same augmentation level, the augmentation is levelled up, i.e. L is increased by one", we did not agree and, instead of following this statement, we increased L whenever the difference between the KID score and the average of the two last ones (attained all in the same augmentation level) was smaller than the 5% of that average. After contacting the authors about this via OpenReview, they agreed with our change.

When the new level is attained, the probability distribution for the new bit of $\boldsymbol{s}$ is not uniform. We use a Bernoulli distribution with

$$p = \min\{0.5(t - t_{st})/t_r, 05\} \tag{1}$$

being the probability of 0. Here t is the current iteration, t_{st} is the last iteration in which $\boldsymbol{s}$ was augmented and t_r is a hyperparameter (10^5 in the experiments). This is justified in the paper saying that it helps with learning stability.

It is also said by the authors that the learning rate of the generator could be decreased when a new level is attained. However, they do not give details about how or where to implement this.Thus, we decided not to implement it but to contact the authors. Once they replied, they confirmed that it was not used by them neither.

Last but not least, in order to evaluate the performance of the GAN we used the Fréchet Inception Distance (FID) as they do in the original paper.

Our PyTorch implementation can be found at `https://github.com/telecombcn-dl/2018-dlai-team1`.

4 EXPERIMENTS

In the original paper, Progressive Augmentation is applied to a Spectral Normalized DCGAN (SN-DCGAN). In order to reproduce the results, we implemented both a simple SN-DCGAN (used as a baseline) and the newly proposed Progressive Augmentation technique. We reproduced the experiments using the same datasets as in (Zhang & Khoreva, 2019), which are MNIST, FashionMNIST, CIFAR10 and CelebA.

For all of the experiments performed we used the same hyperparameters as in the original paper: we used ADAM optimizer with learning rate of 10^{-3} for MNIST and $2 \cdot 10^{-4}$ for the rest. The rest of hyperparameters of ADAM were $\beta_1 = 0.5$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$.

As for the progression scheduling described in section 3, t_r (from equation 1) was fixed to $5 \cdot 10^4$, and the KID was evaluated every 10^5 steps with 10^5 fake samples and 10^5 real samples. One update of the generator was done for every discriminator update. We tried progressive augmentation starting both at level $L = 0$ and $L = 2$. We used the Fréchet Inception Distance (FID) to evaluate the performance of the GAN. A lower FID means that the generated images and their variance are more "real-like".

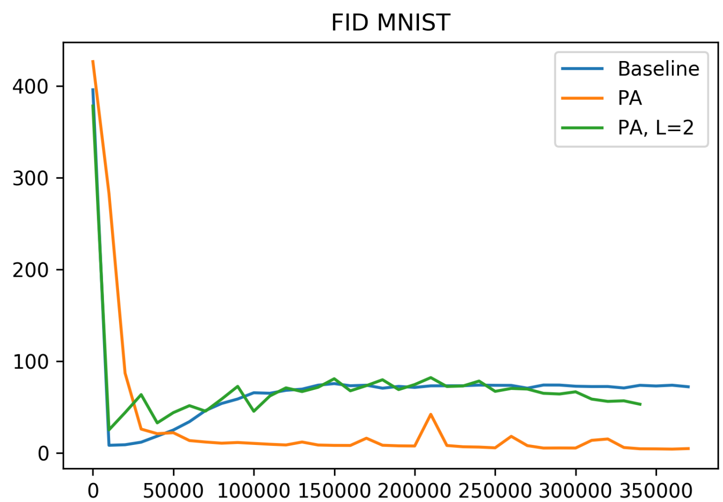


Figure 3: FID value during training with MNIST

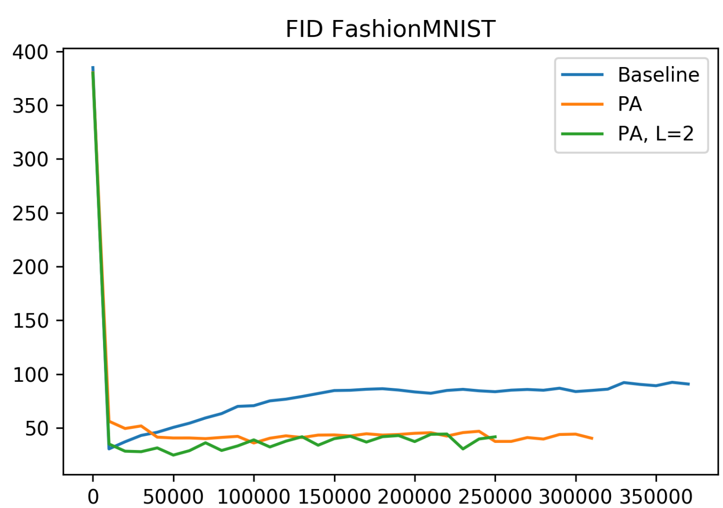


Figure 4: FID value during training with FashionMNIST

In figures 3 and 4, corresponding to MNIST and FashionMNIST we can observe how applying Progressive Augmentation does improve stability. We also observe it helps achieve a lower FID value. On the other hand, in figures 5 and 6 we see that applying progressive augmentation does not seem to increase stability in our experiments, though it helps achieve a lower total FID value anyway.

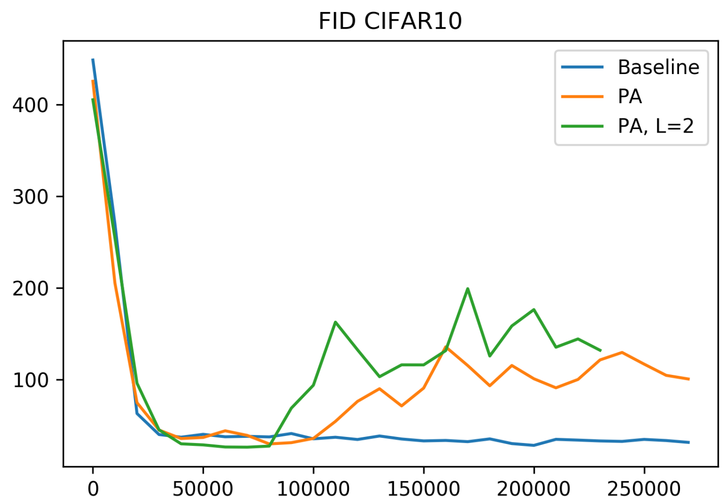


Figure 5: FID value during training with CIFAR10

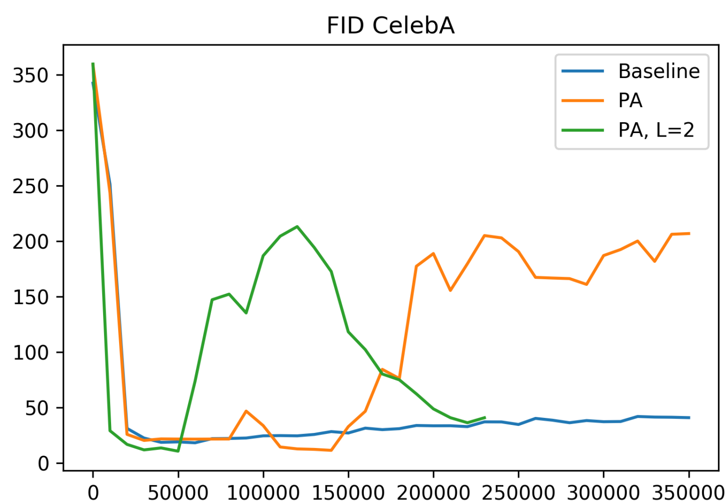


Figure 6: FID value during training with CelebA

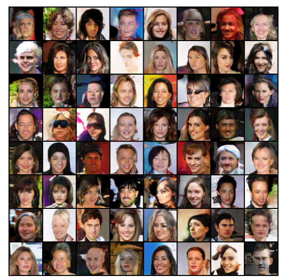
We also show the comparison between the generated images with and without Progressive Augmentation for MNIST (as an example where the difference is clearly appreciable) and CelebA (where the difference is more subtle) in figures 7 and 8.

(a) Baseline

(b) Progressive augmentation

Figure 7: Comparison of training a SN-DCGAN on MNIST with and without Progressive Augmentation for the same number of iterations. Here we can see that PA helps avoid mode collapse.

(a) Baseline



(b) Progressive augmentation

Figure 8: Comparison of training a SN-DCGAN on CelebA with and without Progressive Augmentation for the same number of iterations. For this dataset, it is harder to appreciate the difference between them.

Even though we could see an improvement when applying Progressive Augmentation to a GAN, we could not achieve the exact same results for all datasets as in the original paper. We suspect this might be due to a lack of resources to do as many trials as necessary, or an error in the hyperparameters.

5 CONCLUSIONS

This report reproduces the results of a paper submitted to the Reproducibility Challenge[3] set up in the 2019 International Conference on Learning Representations (ICLR).

The paper chosen in order to reproduce its results was "PA-GAN"[4], which proposed a novel method based on Progressive Augmentation(PA) in order to improve the performance of a GAN by hindering the task of the Discriminator.

We implemented a SN-DCGAN and trained it with 4 different datasets in order to achieve to reproduce a baseline. Besides, we implemented the Progressive Augmentation technique which brings a novel approach to increase GAN's performance. In order to compare them, we used the KID and FID scores, which are metrics used to evaluate GANs. Finally, we were able to reproduce some of the original paper's results but we also found some sections of the paper that, in our opinion, needed more clarification. We contacted the authors regarding those via OpenReview and they were really helpful thus providing details in order to correct the explanation of the scheduling which we pointed out.

To sum up, we could implement the paper partly successfully, and applying Progressive Augmentation seems to improve a simple GAN's performance, as it was claimed originally. However, we

[3]https://reproducibility-challenge.github.io/iclr_2019/

[4]https://openreview.net/pdf?id=ByeNFoRcK7

could not achieve the exact same metrics as in the paper, either for the few trials computed (in comparison with theirs) or for the missclarification when choosing hyperparameters. Some incorrect descriptions were also found in the paper but most of them were solved after contacting the authors. So, even we believe that the results are reliable and reproducible, we also believe that more specificity regarding the hyperparameters used would be helpful for other researchers trying to apply the novel Progressive Augmentation technique.

ACKNOWLEDGMENTS

We would like to especially thank the support team at Xavier Giró and Noe Casas for guiding our project. We gratefully acknowledge the support of Google Cloud for the computational resources.

REFERENCES

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 27*. 2014.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Dan Zhang and Anna Khoreva. PA-GAN: Improving GAN training by progressive augmentation, 2019.