REVIEW OF 'PROGRESSIVE GROWING OF GANS FOR IMPROVED QUALITY, STABILITY, AND VARIATION'

Jay Santokhi, Adrian Wilczynski & Percy Jacks
Electronics and Computer Science
University of Southampton
Southampton, S017 1BJ, UK
{jks1g15,aw11g15,pj2g15}@ecs.soton.ac.uk

ABSTRACT

Karras et al. (2018) introduced a new methodology for training GANs through progressively growing the Generator and Discriminator starting at a low resolution and gradually increasing, through the addition of new network layers. Their work on Progressively Grown GANs (PGGAN) with the addition of two normalisation techniques (Equalised Learning Rate and Pixelwise Feature Vector Normalisation) claimed to improve stability, reduce training time and improve generated image quality over current state of the art GANs. This review paper has found that this methodology and its claimed benefits are mostly valid, however, as well as requiring considerable computational assets it is also ineffective for simple datasets.

1 INTRODUCTION

Generative Adversarial Networks (GANs) first described by Goodfellow et al. (2014) can be used to generate images indistinguishable from their training set images. This framework involves the simultaneous training of two network models: a Generator, G and a Discriminator, D. G captures the distribution of the training set and in turn will accept an input vector of random noise which produces an output that is similar to the original training set. D on the other hand determines whether a given image is from the original training set ('real') or an output of the Generator ('fake').

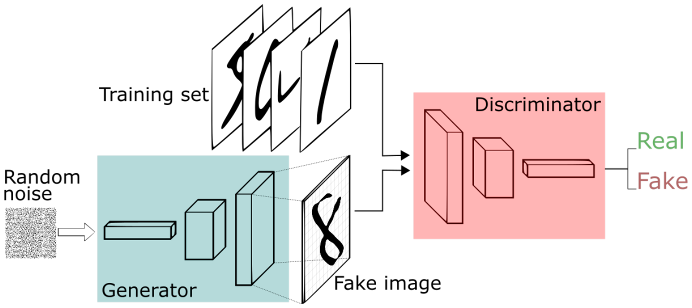


Figure 1: General GAN framework (Silva, 2018)

GANs are inherently difficult to train, the idea being to strike an equilibrium between the loss functions of the two networks during training. If an equilibrium is not met or maintained it can result in a phenomenon referred to as 'Mode/Modal Collapse'. This is the scenario where G will produce the same (or almost same) image and is able to 'fool' D. Mode Collapse starts when D overshoots, leading to exaggerated gradients and an unhealthy competition ensues where signal magnitudes escalate in both networks (Karras et al., 2018). Mode Collapse occurs unpredictably making it difficult to train and evaluate GANs effectively.

This review paper investigates the effectiveness of PGGANs, a new GAN training procedure developed by Karras et al. (2018), against current state of the art guidelines (Radford et al., 2016; Chollet, 2017) and architectures: Deep Convolutional GAN (DCGAN) (Radford et al., 2016) and Least Square GAN (LSGAN) (Mao et al., 2017).

2 Analysis of Chosen Paper

The key idea of Karras et al. (2018) new methodology is to grow both G and D progressively starting from a low resolution which models broad details and adding new layers that model fine details as training progresses. This incremental nature allows the training to first discover large scale features (low frequencies) of the image distribution and then shift attention to finer details (high frequencies) instead of learning all scales simultaneously like current GAN architectures. G and D therefore grow in synchrony with all layers in both networks remaining trainable throughout the training process. When adding new layers, they must be faded in smoothly to avoiding the sudden 'shock' to an already well trained smaller network, see Figure 2. Using this, Karras et al. (2018) created a high quality version of the CELEBA dataset with a resolution higher than that of the original dataset.

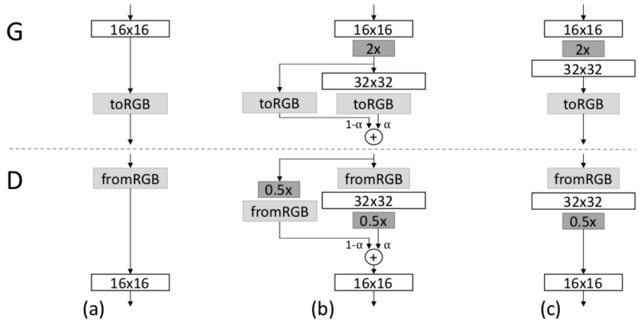


Figure 2: Progressive growing nature of PGGAN. During the transition (b) layers operating with higher resolution are treated like residual blocks whose weighting, α increases linearly from 0 to 1 (Karras et al., 2018).

The authors say their work is independent of loss function. Their experiments mainly used Wasserstein Loss with Gradient Penalty (Gulrajani et al., 2017) but they also experimented with Least-Squares Loss. Generated images appear better with WGAN-GP (which is the 'better' loss function).

A lot of the network structure and the cyclic training nature of PGGAN is assumed; Karras et al. (2018) does not describe how the fading works, only that it needs to be done to avoid sudden shocks. This intuitively makes sense but there is no body of work they point to that suggests why you would want to do this and they do not indicate whether they tried different methods. The impact of the progressive training on mode collapse is also never touched upon.

2.1 Minibatch Standard Deviation (MSD)

While D looks at individual training images and their distributions, G is forced to generate the same kind of variation in results. GANs seem to have a tendency to only capture a subset of this variation. The goal is to encourage minibatches of generated and training images to show similar statistics. Karras et al. (2018) suggest MSD to simplify this approach and improve variation; compute the standard deviation for each feature in each spatial location over the minibatch, average estimates over all features and spatial locations to arrive at a single value, and finally replicate the value and concatenate it to all spatial locations and over the minibatch yielding one additional feature map.

2.2 Equalised Learning Rate (ELR) & Pixelwise Vector Normalisation (PVN)

For normalisation in G and D most solutions involve Batch-Norm in G and often in D, with the intuition being to constrain signal magnitude and competition. Karras et al. (2018) provides a different approach with two methods, neither of which include learnable parameters.

ELR uses $\mathcal{N}(0,1)$ initialisation and explicitly scales weights at runtime. Set $\hat{\omega}_i = \frac{\omega_i}{c}$ where ω_i are weights and c is the per-layer normalisation from He's Initialiser (He, 2015). Karras et al. (2018) says learning rates can be too large and small at the same time, the ELR approach ensures a dynamic range and that learning speeds are the same for all weights making sure all layers learn at the same speed which allows for fair competition between the two networks.

PVN can prevent scenarios leading to mode collapse through normalising the feature vector in each pixel to unit length in G after each convolution layer. This PVN results in G being discouraged from generating images that are 'broken', prevents training from spiralling out of control and prevents feature map magnitudes from getting too large.

3 Experimental Approach

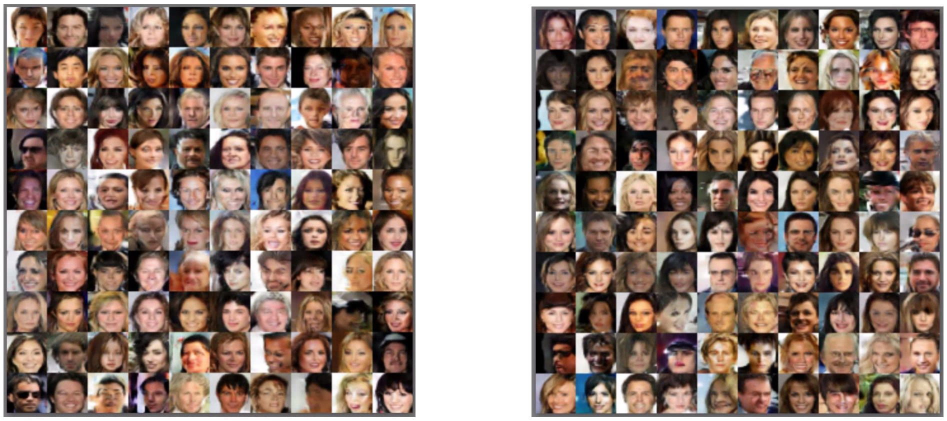
The paper by Karras et al. (2018) described many results and findings however this paper only focuses on the new progressive training procedure, normalisation techniques and MSD. Karras et al. (2018) trained on 8 Tesla V100 GPUs for 4 days to generate images with larger resolutions than that of the original dataset but in order to provide fair and reliable comparisons between different GANs, the PGGAN created here was trained on a single GTX1060 GPU (plus Google Colaboratory) and only generates results up to the resolution of 32x32. Using current state of the art training guidelines from Radford et al. (2016); Chollet (2017) and different GAN architectures with various different datasets (MNIST, MNIST Fashion, CIFAR10 and CELEBA) comparisons of general image quality, time to train and tendency to Modal Collapse were made.

The Radford et al. (2016) architecture guidelines include: replacing pooling layers with strided convolutions "allowing the network to learn its own spatial downsampling", use Batch-Normalisation in both G and D, remove fully-connected (FC) layers in deeper networks, use ReLU activation in G except for the final layer which uses tanh and finally to use Leaky ReLU in D.

The Francois Chollet (2017) training guidelines include: sample random vectors from a normal distribution (i.e Gaussian) rather than a uniform distribution, add Dropout to D, use kernel sizes divisible by stride, and if G loss rises dramatically while D loss falls to zero, reduce learning rate and increase dropout of D.

Models were created using Tensorflow and Keras and the Github Repository for this work can be found at ***https://github.com/COMP6248-Reproducability-Challenge/ReviewOfPGGANs***.

4 Results and Discussion

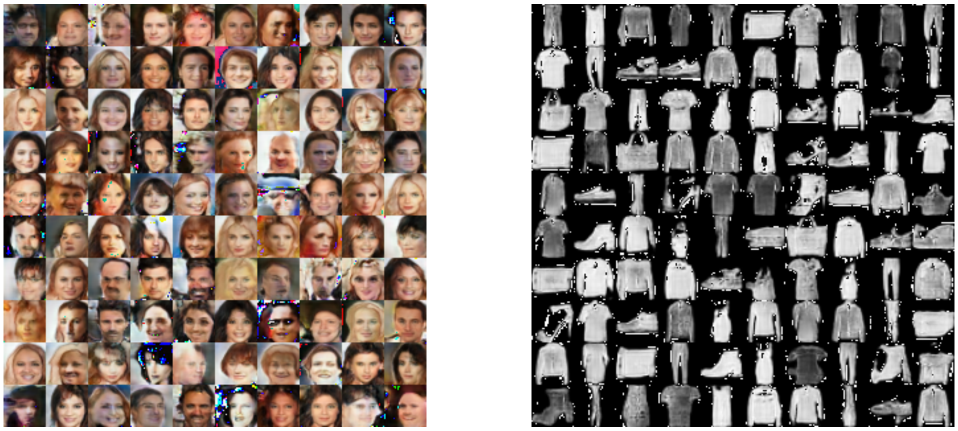


(a) Faces generated using DCGAN

(b) Faces generated using LSGAN

Figure 3: Selected results from using DCGAN and LSGAN

One of the bold claims from Karras et al. (2018) was that their methodology would prevent mode collapse; a claim that was not entirely true. Mode collapse still occurred however PGGAN was able to recover to the result seen in Figure 4a. LS and DCGANs were unable to recover from mode collapse once it started. When comparing the results from LS and DCGAN (Figure 3) with PGGAN, it can be seen that the faces generated using PGGAN are more well formed even though it trained using a smaller latent noise vector (64 rather than 200) as well as considerably less data and time, thus backing up the claims of 'improved quality' and 'reduced training times'. A direct comparison of exact training time would prove fruitless as the PGGAN needs a different number of epochs for each growth cycle. Details of hyperparameters are listed at the Github repository linked.



(a) Faces generated using PGGAN

(b) Fashion items generated using PGGAN

Figure 4: Selected results from using PGGAN to generated images

One unexpected discovery was that despite producing well formed faces, PGGANs were not as effective as traditional GANs for simple datasets such as MNIST and Fashion MNIST. These datasets only contain broad details and thus have no need for a network to 'be careful' when modelling finer details, meaning the intricacies of the PGGAN methodology are wasted. This attempt to model finer details where none exist results in image artefacts akin to salt and pepper noise appearing around the image (see Figure 4b). This cannot be considered mode collapse as outputs and losses are stable, but more of a *Network Apophenia*; where the network is seeking meaning (in the form of finer details) where none exists.

5 Conclusion

In conclusion it is felt that the PGGAN approach described by Karras et al. (2018) is the next phase in the evolution of training GANs to produce unprecedented quality in results, however their inherent complex nature is 'overkill' for simple datasets leading to a condition appropriately coined *Network Apophenia*. In terms of the reproducible nature of the paper, it proved to be very challenging even with a good understanding of all the concepts described. After unsuccessful attempts to implement in Keras, raw Tensorflow was used taking inspiration from various sources (referenced in the Github repository) to develop a PGGAN. On the other hand, development of the traditional GANs using Keras was successful, producing an array of varying results, all viewable at the given Github link.

References

Francois Chollet. *Deep Learning with Python*. 2017.

Ian J Goodfellow, Jean Pouget-abadie, Mehdi Mirza, Bing Xu, and David Warde-farley. Generative Adversarial Nets. pp. 1–9, 2014.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved Training of Wasserstein GANs. 2017.

Kaiming He. Delving Deep into Rectifiers : Surpassing Human-Level Performance on ImageNet Classification. 2015.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANS for Improved Quality, Stability, and Variation. *ICLR*, pp. 1–26, 2018.

Xudong Mao, Qing Li, Haoran Xie, Raymond Y K Lau, Zhen Wang, and Stephen Paul Smolley. Least Squares Generative Adversarial Networks. pp. 1–16, 2017.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *ICLR*, pp. 1–16, 2016.

Thalles Silva. An intuitive introduction to Generative Adversarial Networks (GANs), 2018.