REPORT OF THE COMP6248 REPRODUCIBILITY CHALLENGE

Xinyi Zhu & Hao Chen & Weiyang Huo
School of Electronics and Computer Science
University of Southampton
University Road, Southampton, SO17 1BJ, United Kingdom
{xz6m18,hc1u18,wh3n18}@soton.ac.uk

ABSTRACT

This report describes the reproduction of a part of the paper written by Brock et al. (2018). It firstly expounds the core ideas and methods of the thesis, then describes the experimental methods and specific details in the process of recurrence, and finally analyzes the performance of the GAN algorithm in the paper. The project was completed by three members in the group.

1 INTRODUCTION

Yann LeCun indicated that GAN is one of the most anticipated algorithms in AI and the most interesting idea in the past decade. GAN consists of a Generator and a Discriminator. The two networks constantly adjust the parameters in the confrontation until the Discriminator cannot judge whether the output of the Generator is true. Therefore, GAN can be used to generate images that the model has never seen before, from the real implementation.

We chose this paper from DeepMind for reproduction because it proposed a new GAN model, BigGAN, which is known as the strongest GAN image generator in history because of its outstanding performance. The innovation of this research is to introduce the idea of orthogonal regularization. The GAN generation performance is greatly improved by timely truncation of the input prior distribution z. Under the ImageNet data set, the Inception Score is more than 100 points higher than the best GAN model SAGAN (Close to 2 times).

The study showed that GAN performed better in large-scale training and was able to train at a rate of 2 to 4 times with the same conditions. The authors used two simple generation architectures that increased scalability and modified the regularization scheme to improve conditioning. The model becomes subject to truncation techniques as a side effect of the modification method. This is a simple sampling technique that allows fine-grained control over sample diversity and fidelity. In addition, the study found the instability of large-scale GAN and described it empirically. It has been shown from this analysis that combining new and existing techniques can reduce this instability, but achieving full training stability must be at the expense of performance. Fig.1 shows the results produced by BigGAN.

Figure 1: results produced by BigGAN

2 Model

2.1 Architecture

The architecture of Big-GAN is shown in figure 2. (a) is the typical architecture of Big-GAN. The network is composed with several (a)s, and (b) and (c) are the details of the ResBlock (He et al. (2016)) in Generator and Discriminator. The latent vector z is split and concatenated to the corresponding output of each ResBlock. Non-local block is attention model method, which will be discussed in section 2.4.

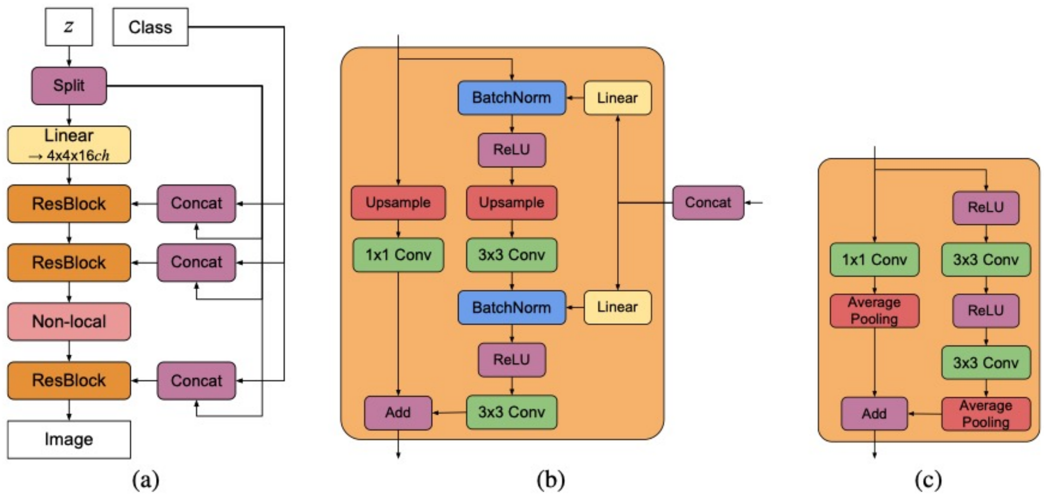

Figure 2: Architecture of Big-GAN

2.2 Scale Up the Model

First, comparing with many previous models, Brock et al. (2018). find that scaling up GAN can help boost the performance, they increase batch size from 256 to 512, to 1024 and to 2048. Bigger batch covers more modes, it not only provides more accurate gradient, but also decrease the training iterations. They also increase the channels from 64 to 96 and find that the improvement is significant.

Brock et al. (2018) also propose three improve methods, the first one is to use shared embedding which use the same linear mapping matrix for every conditional batch normalization layer to speed up the training. The second one is use orthogonal regularization which is used to help avoid distribution shift when using truncation distribution as the latent space. The third one is use hierarchy latent vector which adding a noise vector z to every residual block (see Fig.2. (a)) to help receive the information from latent space.

2.3 Truncation Trick

Truncation trick is proposed as a sampling method to control fidelity and variance. The latent vector z is arbitrarily sampled from a latent space. Vast majority choose $N(0, I)$ or $U[-1, 1]$ as the latent space. Rather than sampling from the entire distribution, truncation trick will resample z if its value above a threshold, which can improve the sample quality at the cost of overall variety.

The bigger truncation threshold is, the more modes latent space can cover, which leads to more variety images; the smaller truncation threshold is, the better the effect is, but the fidelity is worse. This truncation trick allowing fine control over the trade-off between sample fidelity and variety by reducing the variance of the G's input. (Brock et al. (2018))

2.4 ATTENTION MODEL(NON-LOCAL)

Big-GAN adds self-attention block to improve the ability of Generator and Discriminator. Self-attention block is first introduced in SAGAN by Zhang et al. (2018). With self-attention block, the model learns to efficiently find global, long-range dependencies within internal representations of images. The basic principle of self-attention block is that given a feature map it does not focus on every pixel on the feature map. Each pixel of the feature map has a weight, which indicate the how much attention it is paid. Instead of one single feature, self-attention block will generate a attention map and a feature map, then pixel-wisely multiply these two maps. The self-attention block structure is shown in the Fig.3 as below:

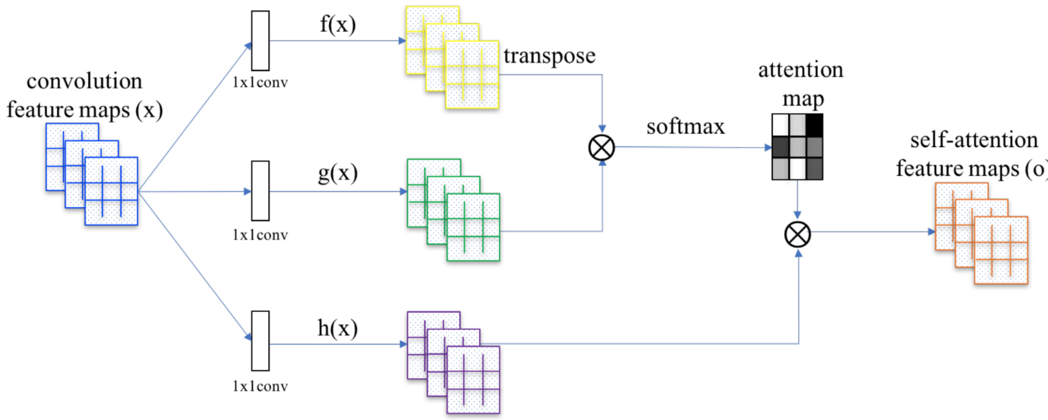

Figure 3: Self-Attention Block introduced by Zhang et al. (2018)

The number of channels in the original feature maps will be reduced through 1x1 convolution. The image features will be then transformed into three feature maps, $\boldsymbol{f}(\boldsymbol{x})$, $\boldsymbol{g}(\boldsymbol{x})$ and $\boldsymbol{h}(\boldsymbol{x})$. Let β denote the attention map, then $\beta = \boldsymbol{f}(\boldsymbol{x})^T\boldsymbol{g}(\boldsymbol{x})$. $\beta_{i,j}$ indicates the extent to which the model pay attention on the i^{th} row and j^{th} column. The final output of self-attention block is $o = \beta\boldsymbol{h}(\boldsymbol{x})$.

3 EXPERIMENT

3.1 REDUCED PARAMETERS AND DEVICES

The author of BigGAN has released a PyTorch version of BigGAN on the Github, the original model was trained with a batch size of 256 and 8 gradient accumulations, for a total batch size of 2048. The pre-trained models were trained on 8xV100 (16 GB VRAM each) with full-precision training and takes 15 days to train to 150k iterations.

Since we only have one GPU on the colab, it is apparent that we need to tune the model a little bit. We reduced the batch size from 256 to 24, num_works from 8 to 2, and change the iteration to 100 and 500 to fit our situation.

3.2 DATASET

We did not use the ImageNet database here, alternatively, we used a small dataset which includes 1087 items of anime avatars. To train the model, we change the path of dataset to $/data/ImageNet/face\_small$ to fit the model's settings.

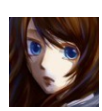

Figure 4: An example of anime avatar dataset.

3.3 RESULTS

(a) (b) (c)

Figure 5: Samples from model with truncation 0.5. (a) Epoch 10. (b) Epoch 100. (c) Epoch 200.

From Fig.5, it is apparent that with only 10 epoch, the result is a mess. In (b), the shape of a head begins to show up, including facial organs. In (c), the effect is much more better. Comparing with the original avatars, from (c) we can find that there are still much noise around the organs, and some are mixture of many different faces.

4 CONCLUSION AND FUTURE WORK

In summary, this BigGAN has a highly performance, according to Brock et al. (2018) . (2018), when they trained on ImageNet at 128x128 resolution, the Inception Score (IS) is 166.5 and Frechet Inception Distance (FID) is 7.4, increasing more than 100 score comparing with previous best IS of 52.52 and FID of 18.65. With scaling up the model (8x batch size, 2x model size), splitting noise into multiple chunks and sampling from truncated distribution space, it is a significant contribution to boost the performance and the result is very impressive. Due to the limitation of the one GPU, we cannot train a full-size BigGAN but a simple version. According to some advises from Brock, we could also reduce the $num\_workers$ of the model and increase $num\_G\_accumulations$ and $num\_G\_accumulations$ to remedy for the reduced batch size, but anyway, a higher compute power will have more significant help.

REFERENCES

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *CoRR*, abs/1809.11096, 2018. URL `http://arxiv.org/abs/1809.11096`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.