Improving cGANs with Parameterized Discriminators

Xu Huirong 20074688D
Supervisor: Prof. Huang Jian

Hong Kong Polytechnic University
Department of Applied Mathematics

December 2023

Abstract

Generative Adversarial Nets(cite Conditional GANs (8)) (GANs) are one power tool of machine learning designed to solve generative problems, which achieves state-of-art performance in generating images. GANs have many advantages such as it can generate high-quality images without incremental generation process(cite GANs). However, it suffers from problems such as there is no way to control on modes of the data that is generated. (cite cGANs) To solve this issue, Mehdi and Simon proposed Conditional Generative Adversarial Nets, which feed the data y to both generator and discriminator to generate images according to specific labels or give multiple tags for pictures in multi-modal learning (cite cGANs).

To further enhance the training of GANs, this project explores details of Auxiliary Classifier GANs (AC-GANs), which shows that adding one auxiliary decoder network to the discriminator could greatly improve the quality of images and also produce high-resolution images(cite AC GANs), results and assessment of picture diversity and discriminability would be discussed.

Apart from AC-GANs, this project investigates projection discriminators in cGANs. Instead of embedding conditional vectors to the feature vectors like the traditional cGANs, it used a parameterized discriminator with an inner product between condition y and the feature vector(cite Projection GANs). I will describe the experimental details and technique of both Projection GANs and AC-GANs, with a thorough comparison providing important information for future research paths.

Acknowledgment

Throughout this project, I have received a great number of supports from the Department of Applied Mathematics. Firstly, I would like to express my gratitude and appreciation to my supervisor, Chair Prof. Huang Jian, for his supervision, patient help, and insightful guidance throughout the duration of the Final Year Project.

Next, I also like to thank my mentor, Ph.D. Niu Shengjie, for his regular support and directions offered for this project. His knowledge was really helpful in getting this project started.

At last, thanks to all the IT staff in PolyU and AMA for their continuous help with my server setup and debugging, without which I could not get the project running.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

Supervised Learning is the most widely used and successful machine learning model for tasks like classification[6]. They learned to map a complex input to some relatively easy output. For example, from images to labels of a dog or cat. Supervised learning is relatively easy and can achieve satisfactory performance after the training process. However, it relies on millions of training inputs and human supervision to perform well[6].

To address those problems, unsupervised learning is introduced; it learns patterns with unlabeled data. Clustering and reduction in dimension are commonly used. Another approach to unsupervised learning is the generative model, which aims to learn a $p_{\text{model}}(x)$ that could assemble $p_{\text{data}}(x)$. Goodfellow et al. (2014)[6] proposed Generative Adversarial Networks (GANs), which utilized generative models to avoid having to deal with the challenge of approximating a lot of complicated probabilistic computations[7]. GANs could generate convincing image samples on datasets with low variability and low resolution[3][11]. The discriminator $D(x)$, which assesses the divergence between the target distribution $q(x)$ and the current generative distribution $p_{\text{G}}(x)$ is the most characteristic feature of GANs[9][1] and will be emphatically introduced in the later parts.

Still, GANs suffer from not being capable of generating high-resolution and coherent pictures. Mirza and Osindero proposed Conditional Generative Adversarial Nets (cGANs)[7], which tried to advance generative models and make them able to generate images or tags according to specific tasks. It has been proven to be a useful tool for tasks like class conditional image generation. Now that there are an increasing number of applications for cGANs. CGANs with auxiliary classifiers[10] introduced some modifications to the latent space to produce high-quality samples and came up with a new matric for accessing varability. CGANs with projection discriminator by Miyato and Koyama[8]are also worth discussing for its parameterized discriminator and different structures from the normal cGANs.

1.2 Objective

1.3 Limitations

1.4 Source of Data

2 Related Work

2.1 Generative models

As a subclass of unsupervised learning, generative models seek to produce new data points with some variability by understanding the training set's true data distribution. These models are frequently utilized in many domains, including speech recognition, natural language processing, and computer vision.

indent?

Variational auto-encoders (VAEs), Generative adversarial Networks (GANs), and autoregressive models are the most widely used generative models. These models have demonstrated success in producing realistic, high-quality samples that are nearly identical to real data.

The two primary types of generative picture models that have been thoroughly examined are non-parametric and parametric. Non-parametric models use training image patches to replicate various functions such as texture generation[4] and super-resolution[5]. Generative parametric models are used in an extensive variety of deep learning techniques. For example, generative decoders are used in restricted Boltzmann machines, and denoising auto-encoders are used to rebuild the image from its latent form.[2]

2.2 Generative Adversarial Nets

Generative Adversarial Nets was first introduced by Goodfellow et al.[6] in 2014. It developed a deep implicit generative model that could produce real samples in a single generation step without relying on the incremental generation process or Markov chains.

no need to textit the second time

The theory behind GANs is game theory. One network is called the *generator*; it has a prior distribution $p(z)$ over a vector z, which is the input to this *generator*. Another input to the *generator function* is the trainable parameter θ, which controls the criteria used in the game. Typically, the prior distribution $p(z)$ is an unstructured distribution, like a uniform distribution or a high-dimensional Gaussian distribution; samples z from this distribution are merely noise[6]. The main role of the *generator* is to transform the mere noise z to realistic examples. Another important component is the *discriminator*. It has a second multilayer perceptron $D(x, \theta)$, which will output a single scaler. The likelihood that x originated from the data instead of p_g is represented by $D(x)$. Theoretically, the GANs is just a minmax game between *generator* and *discriminator*. The *generator* tries to produce samples that the *discriminator* cannot tell if they're from the real world or from the *generator*. On the other hand, the *discriminator* makes efforts to distinguish between generated samples and real samples; it is trained to assign both generated labels and real labels correctly to the two clusters, while *generator* minimizes $\log(1 - D(G(z)))$. They play against each other to obtain the following minmax function:

add '.'

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

High-quality, realistic data is produced by the generator through this adversarial process. GANs have proven effective in several applications, such as image synthesis, super-resolution, and style transfer.

2.3 Conditional Generative Adversarial Nets

Conditional Generative Adversarial Nets was introduced by Mirza and Osindero in 2014[7]. It is construced by feeding the data with condition y. y could be class labels or information from other modalities, and y is both construced on *generator* and *discriminator*. The minmax function of Conditional Generative Adversarial Nets can be expressed as:

$$\min_G \max_D \mathbb{E}_{(x,y) \sim p_{\text{data}}(x,y)}[\log D(x, y)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z, y)))] \tag{2}$$

The mode collapse issue that frequently arises with GANs is resolved by this conditioning. It also enables the model to produce data according to specific tasks or labels. cGANs have been applied to a number of tasks, such as attribute-based face generation, text-to-image synthesis, and image-to-image translation.

3 Project Schedule

Stages	Tasks	Time Allocation
Research and topic choosing	– Discuss with supervisor, searching for the best topic. – Get fundamental knowledge about GANs and its application. – Identify necessary datasets, software which will be used for this topic.	2 months
Setting up software and access GPU	– Contact IT staff in AMA, require GPU access. – Install and configure the necessary packages and set up the environment.	1 month
Interim report writing	– Select and read papers about ACGANs and Projection GANs, understanding their methodology. – Confirm the structure of the report and parts that should be finished in the interim report. – Summarize and compare, write the interim report.	1-2 months
Code implementation	– Training and testing on existing code in ACGANs and Projection GANs. – Fine-tuning and modification on existing code. – Refine and optimize existing methods. – Compare and document the result.	1-2 months
Final reporting	– Finish the final report, summarizing all the works and outcomes. – Make use of more related materials to enrich the reports and dive deeper into the discussion.	1 month

Table 1: Project Schedule

4 Resource Utilization

4.1 Hardware Resources

The project used the GPUs from Prof. Huang Jian's GPU workstation, which has four A100 to use from. Also, in order to access the server outside campus, I applied for research VPN.

4.2 Software Resources

Python was the main language chosen for its relative easiness to use and its robust performance.

5 Methodology

5.1 Overview

5.2 ACGANs

5.3 Projection GANs

6 Results and Findings

7 Reference

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, pages 214–223, 2017.

[2] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. *arXiv preprint arXiv:1506.05751*, 2015.

[3] Emily L. Denton, Soumith Chintala, Arthur Szlam, and Robert Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. *CoRR*, 2015.

[4] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *ICCV*, volume 2, pages 1033–1038. IEEE, 1999.

[5] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *Computer Graphics and Applications, IEEE*, 22(2):56–65, 2002.

[6] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[7] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[8] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. In *ICLR*, 2018.

[9] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *NIPS*, pages 271–279, 2016.

[10] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, pages 2642–2651, 2017.

[11] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, 2015.