Analysis of GANs approaches on MNIST dataset

Harikrishnan Seetharaman, Jemuel Stanley Premkumar
hariksee@umich.edu, jemprem@umich.edu

I. Motivation

Generative Adversarial Networks (GANs) is one of the latest and most powerful areas of unsupervised learning and generative modeling. A typical GAN has two types of neural network inspired from the game theory, one network known as the generator G that generates data based on a model it has created using samples of real data received as input. The generator learns the underlying structure by using a number of parameters significantly smaller than the amount of data it has trained on, which is a core objective of unsupervised learning strategy. The other network known as the discriminator D discriminates between data created by the generator and data from true distribution. The two networks are locked in a zero-sum game where the generator is trying to fool the discriminator into thinking that the synthetic data comes from the true distribution, and the discriminator is trying to call out the synthetic data as fake.

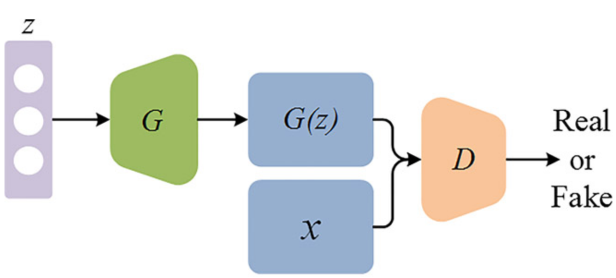

Fig. 1: Generative Adversarial Modeling

GANs provide a path to sophisticated domain-specific data augmentation and a solution to problems that require a generative solution, such as image-to-image translation. Implementing various GANs would allow us to understand the popular approaches practised in generative modeling. From a theoretical perspective, this would not only allow one to understand different losses and evaluation metrics but also understand the underlying math corresponding to latent spaces.

Coming from a robotics background, this project would allow us to develop models for applications such as generating grasp rectangle using Pix2Pix [1] or for generating depth image from RGB image directly as most of the 6-DOF grasping policies require both the RGB image and its corresponding depth information[2].

II. Supporting Materials

The Generative Adversarial Networks with Python[3] would serve as our primary reference throughout the project. The models would be trained on the MNIST dataset[4]. The official TensorFlow framework and documentation will be used to implement the different architectures on Python. These papers[5][6] would be used to implement various evaluation metrics; to name a few: Inception score, Frechet-Inception distance, Kernel-Inception distance.

III. Plans

1) We plan to train the following GANs architectures in various domains, classified as below using tensorflow-keras and numpy framework:
   a) Base: Deep Convolutional Generative Adversarial Network (DCGAN) [7]
   b) Structural modifications
      i) Least Squares Generative Adversarial Network (LSGAN) [8]
      ii) Wasserstein Generative Adversarial Network (WGAN). [9]
   c) Image-to-Image transformation
      i) Pix2Pix
      ii) Cycle GAN
   d) Conditioned
      i) InfoGAN
      ii) StyleGAN
2) Implement appropriate evaluation metrics
3) Draw inference from the loss and convergence curves
4) Compare and document the performance of different models on MNIST dataset

REFERENCES

[1] V. Kushwaha, P. Shukla, and G. C. Nandi, "Generating quality grasp rectangle using pix2pix gan for intelligent robot grasping," 2022. [Online]. Available: https://arxiv.org/abs/2202.09821

[2] L. Chen, S. Lin, Y. Xie, Y. Lin, W. Fan, and X. Xie, "DGGAN: depth-image guided generative adversarial networks for disentangling RGB and depth images in 3d hand pose estimation," *CoRR*, vol. abs/2012.03197, 2020. [Online]. Available: https://arxiv.org/abs/2012.03197

[3] J. Brownlee, *Generative Adversarial Networks with Python: Deep Learning Generative Models for Image Synthesis and Image Translation*. Machine Learning Mastery, 2019. [Online]. Available: https://books.google.com/books?id=YBimDwAAQBAJ

[4] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, vol. 2, 2010.

[5] Q. Xu, G. Huang, Y. Yuan, C. Guo, Y. Sun, F. Wu, and K. Q. Weinberger, "An empirical study on evaluation metrics of generative adversarial networks," *CoRR*, vol. abs/1806.07755, 2018. [Online]. Available: http://arxiv.org/abs/1806.07755

[6] A. Borji, "Pros and cons of GAN evaluation measures: New developments," *CoRR*, vol. abs/2103.09396, 2021. [Online]. Available: https://arxiv.org/abs/2103.09396

[7] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015. [Online]. Available: https://arxiv.org/abs/1511.06434

[8] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, and Z. Wang, "Multi-class generative adversarial networks with the L2 loss function," *CoRR*, vol. abs/1611.04076, 2016. [Online]. Available: http://arxiv.org/abs/1611.04076

[9] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," 2017. [Online]. Available: https://arxiv.org/abs/1701.07875