Project Proposal - ECE 285

Sipeng Zhang
Department of Electrical and Computer Engineering
A69027503

Chunlin Chen
Department of Electrical and Computer Engineering
A59023021

Abstract

In this project, our main goal is to reimplement the Deep Convolutional Generative Adversarial Networks (DCGANs)[1] and test its performance on STL-10 dataset[2]. We are going to implement the network from scratch and test its generative ability. DGGANs is a variant of GANs[3], which utilizes convolutional layers to recover image from noise. It follows the working paradigm of GANs, which is to train a generative and discriminative model simultaneously. We would slao like to explore and reimplement more advanced GAN-based architectures if time allows, such as StyleGAN[4] and CycleGAN[5].

1 Problem Definition

we are interested in generative models because we are not familiar with them and would like to gain hands-on experiences in training and tuning these models. Also, it is always motivating to see our own implementation to work.

The generative model paradigm follows GANs is straightforward: to train a generator and a discriminator. The role of discriminator is to help the generator generate image with high quality. One important thing in the training is set the loss function to minimize probability of which generated image is from the original dataset while maxmize probability of which generated image is generated.

In this project, we need to switch to another dataset which is not mentioned in the original paper. For both simplicity and comprehensiveness, we would like to choose STL-10 dataset, which contains 100,000 unlabeled images. Also, it is partly because a too small dataset would just have bad generated results.

2 Tentative Method

There are maily 5 improvements mentioned in DCGAN to improve the stability of GAN[1].

- Replace any pooling layere with strided convolutions(discriminator) and fractional-strided convolutions(generator).
- Use batchnorm in both the generator and the discriminator.
- Remove fully connected hidden layers for deeper architectures.
- Use ReLU activation in generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

In our implementation, we would pay special attention to setting above and maybe further do abalation study on how these setting affect the final result.

3 Experiments

Our main dataset in this project is STL-10. The STL-10 dataset is an image recognition dataset for developing unsupervised feature learning, deep learning, self-taught learning algorithms. It is inspired by the CIFAR-10 dataset but with some modifications. In particular, each class has fewer labeled training examples than in CIFAR-10, but a very large set of unlabeled examples is provided to learn image models prior to supervised training. The primary challenge is to make use of the unlabeled data (which comes from a similar but different distribution from the labeled data) to build a useful prior. The image in this dataset is 96x96, RGB channel.

We would first train the DCGAN on STL-10 dataset and test its generative ability by showing the generated image and check its quality. Also, we would like to

- the datasets you are planning to use:
  the brief introduction of the dataset
  the data format
  other information related to your experiments
- the experiments are planning to perform, and the purpose of performing it.

References

[1] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[2] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 215–223.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[4] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[5] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.