Project Proposal - ECE 285

Haonan Peng
Department of ECE
A14765890

LinXiao Zhang
Department of ECE
A59013665

Abstract

DCGAN is one of the popular and successful network design for GAN. It mainly composes of convolution layers without max pooling or fully connected layers. It uses convolutional stride and transposed convolution for the downsampling and the upsampling. In this project we want to reimplement the DCGAN model and improve it by trying different activation functions and network architectures. Experimental results on CelebA show that our model performs better than the original DCGAN. Besides, we conducted a series of ablation studies to explore the impact of individual components.

1 Introduction

DCGAN is the standard convolutional baseline that many GAN architectures were based upon, it can generate high quality images by using strided convolutional layers in the discriminator to downsample the images and fractionally-strided convolutional layers to unsample the images. The importance of DCGAN is that it contributes significantly to balancing GAN training with its convolutional architecture, GAN and naturally DCGAN have an unsupervised network structure. In order to better understand how DCGAN works we plan to reimplement the DCGAN model and train it on a new dataset to generate new images of fake human faces. Besides, we improved the original DCGAN by changing its activation function and conducted ablation studies to access the impact of individual components.

2 Related work

Generative adversarial network: A generative adversarial network (GAN) is a class of machine learning frameworks designed by Ian Goodfellow and his colleagues in June 2014[1]. Two neural networks contest with each other in a game (in the form of a zero-sum game, where one agent's gain is another agent's loss). Given a training set, this technique learns to generate new data with the same statistics as the training set. For example, a GAN trained on photographs can generate new photographs that look at least superficially authentic to human observers, having many realistic characteristics. Though originally proposed as a form of generative model for unsupervised learning, GANs have also proved useful for semi-supervised learning, fully supervised learning and reinforcement learning. The core idea of a GAN is based on the "indirect" training through the discriminator, another neural network that is able to tell how much an input is "realistic", which itself is also being updated dynamically. This basically means that the generator is not trained to minimize the distance to a specific image, but rather to fool the discriminator. This enables the model to learn in an unsupervised manner.

Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks: A deep convolutional generative adversarial network(DCGAN) is a direct extension of the GAN, except that it explicitly uses convolutional and convolutional-transpose layers in the discriminator and generator, respectively. It was first described by Radford et. al. in the paper Unsupervised Representation Learning With Deep Convolutional Generative Adversarial Networks[2]. The discrim-

inator is made up of strided convolution layers, batch norm layers, and LeakyReLU activations. The generator is comprised of convolutional-transpose layers, batch norm layers, and ReLU activations.

3 Dataset

We are using the CelebA dataset. CelebFaces Attributes Dataset (CelebA) is a large-scale face attributes dataset, which contains more than 200K celebrity images, each with 40 attribute annotations, for non-commercial research purposes. The images in this dataset cover large pose variations and background clutter. CelebA has large diversities, large quantities, and rich annotations, including 10,177 number of identities, 202,599 number of face images and 5 landmark locations, 40 binary attributes annotations per image. The dataset can be employed as the training and test sets for the following computer vision tasks: face attribute recognition, face recognition, face detection, landmark (or facial part) localization, and face editing and synthesis.

4 Methodology

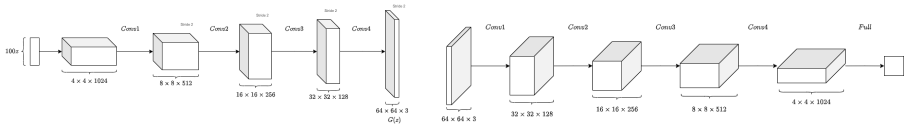
As we all know, the DCGAN is a direct extension of the GAN but implemented with convolutional and convolutional-transpose layers in both the discriminator and generator. According to Goodfellow's paper[1], the GANs are combined with two different models: generator and discriminator. In general, the goal of generator is to self-generate "fake" images with respect to the natural images used for training, while the discriminator is responsible for classifying input image whether it is spawned from generator or came from the natural. During the training process, the generator continuously produces deceptive images to misguide the discriminator and improves the quality of images from that. Meanwhile, the discriminator is trained to be as vigilant to the deceptive input from the generator as possible. Then the equilibrium of this competition is that the discriminator can only guess at 50% confidence whether the input is real from the training set, or it is perfectly faked by the generator.

In mathematical representation, if we denote the input data of an image as x, and $D(x)$ is the output from the discriminator indicates the probability that x is real data rather than fakes from the generator. It means that $D(x)$ is intuitively high if x is from training data while it is low when x is from the generator. As for the generator, let z be a latent space vector sample formed by standard normal distribution, $G(z)$ is the mapping of z from latent space to data space, i.e., the image with HWC size $64 \times 64 \times 3$. G is the differentiable function that estimates the distribution of training data from p_{data} to generate fake samples from that distribution p_g. $D(G(z))$ indicates the probability that output from generator is classified as real image. Finally, the minimax competition of two players can be mathematical described as[1]:

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

Generator: In our project, the generator is first composed of several stride two dimensional convolutional transpose layers in series, while each layer is paired with a 2D batch normalization layer and a $ReLU$ activation layer to adjust the flow of gradient during training. Then the output is fed into the $tanh$ function after passing the last convolutional transpose layer that contains no batch norm layer and activation to turn its range to $[-1, 1]$. With this architecture, the generator take latent space vector with length 100 and generate an output image with HWC size $64 \times 64 \times 3$.

Discriminator:For discriminator, its architecture is like the reversed version of generator, which means that it takes image as input and outputs the probability that the input is from real training data rather than the generator. It is also composed of several stride two dimensional convolutional transpose layers in series, but each layer is paired with 2D batch norm layer and leaky $ReLU$ activation rather than common $ReLU$. Similar to the paired layers after strided convolutional layers in generator, the paired batch norm layer and leaky $ReLU$ activation keep the gradient flow healthy during training.



(a) generator (b) generator

Figure 1: DCGAN architecture

5 Experiments and Results

Preprocess In this project, we use the preprocessed images that are first aligned by similarity transformation with respect to the locations of two eyes, and then their sizes are cropped into 218*178. Since the total number of images in dataset is more than 200k that may bring unignorable burdens on initialization and model training, raising up the expense of later ablation experiments and a well-trained model for final demonstration, we write a script to extract images from the original dataset by a specific attribute according to the provided list of attributes. On the other hand, we believe that maintaining a considerable diversity of attributes ought to help the generator fake more reasonable images to earn better score from the discriminator. Based on that assumption, it should be careful to choose an attribute for images extraction, which means the chosen attribute ought to be as gender free as possible, in our project. To be more specific, attributes such as male, mustache, and even bald, should always not be considered since they intuitively have strong preference on gender leading a lower down on diversity. Therefore, we choose attribute No. 40 which means young people for images extraction which returns a dataset with about half number of images as the original one.

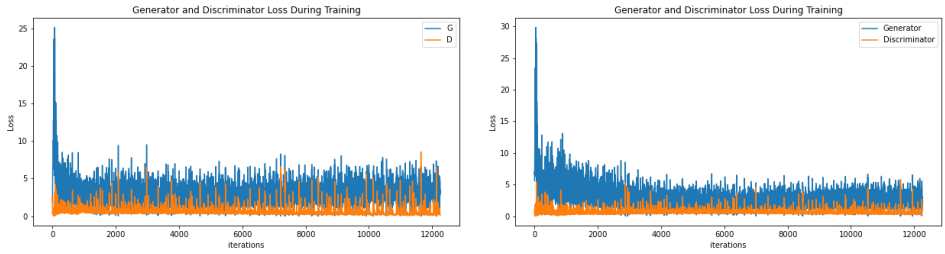
Experiment As mentioned above, we tried different activation functions and conducted a series of ablation experiments. We replaced ReLU with LeakyReLU and found that the model with LeakyReLU performs better than the model with ReLU. For LeakyReLU, we tried different slope, the model with LeakyReLu can achieve the best performance when slope equals to 2. Figure 2 shows the pictures generated by the model with ReLU and the model with LeakyReLU, Figure 3 shows the loss of the two models. From these two figures we can see that the model with LeakyReLU performs better.

The DCGAN architecture is comprised of two models: a discriminator and a generator. The discriminator is trained directly on real and generated images and is responsible for classifying images as real or fake (generated). The generator is not trained directly and instead is trained via the discriminator model. Figure 3 shows that for the model with LeakyReLU, the loss of discriminator and generator are closer, which can help the training of the discriminator and generator.

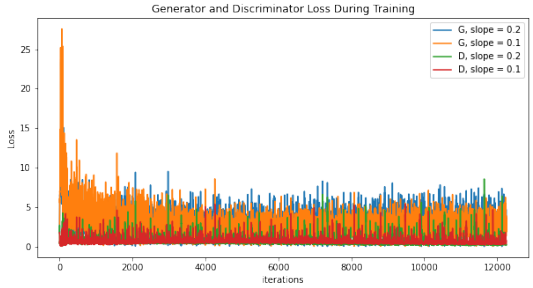
We also tried different slope for LeakyReLU, Figure 4 shows the loss of both Generator and Discriminator with different slope values.

(a) ReLU vs. Leaky ReLU, slope = 0.2) (b) Leaky ReLUs with different slopes

Figure 2: pictures generated by the models with ReLU and LeakyReLU



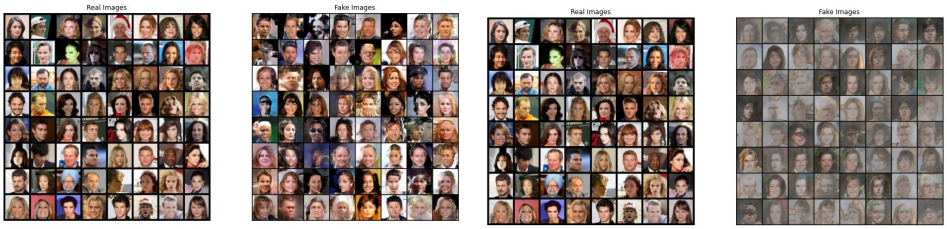
(a) ReLU (b) Leaky ReLU, slope = 0.2



(c) Leaky ReLUs with different slopes

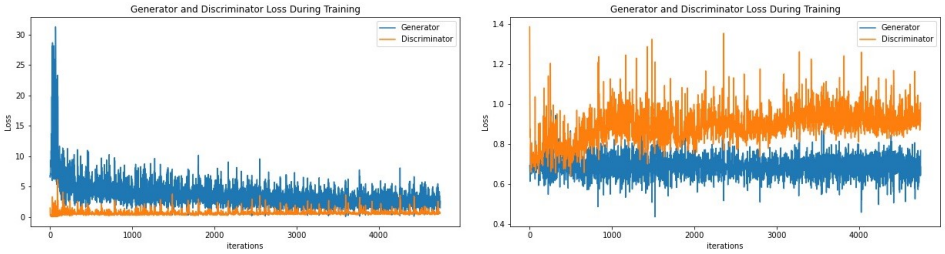
Figure 3: loss of the models with ReLU and LeakyReLU

The first ablation study is to see the effect of batchnorm. From the experiment result below we can see that the pictures generated by the model without batchnorm are dimmer. Batch Normalization can reduce the dependence of gradients on the scale of the parameters or their initial values, regularize the model and reduce the need for dropout, photometric distortions, local response normalization and other regularization techniques. Figure 5 shows the pictures generated by the two models, figure 6 shows the loss of the two models. From these two figures we can see that the performance of the model with batchnorm is better and its loss is smaller.



(a) with batchnorm (b) without batchnorm

Figure 4: pictures generated from batch normalization ablation experiment

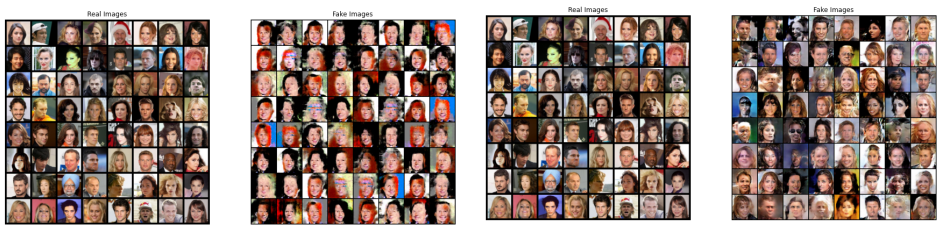


(a) with batchnorm (b) without batchnorm

Figure 5: the loss of the models with batchnorm and without batchnorm

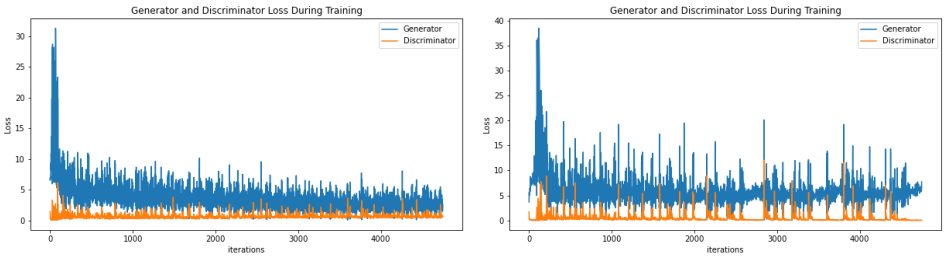
The second ablation study is to see the effect of dropout. The result shows that the model without dropout performs better. Dropout is generally less effective when regularizing convolutional layers, the reason is that convolutional layers have few parameters, so they initially require fewer regularization operations. Besides, due to the spatial relationship of feature maps, activation values can become highly correlated, which makes Dropout ineffective. Figure 7 and Figure 8 show the result of the ablation study of dropout.

We also tested different dropout values and found that the DCGAN model can achieve better performance with smaller dropout valuea. Figure 9 shows the pictures generated by the model with different dropout values and Figure 10 shows the loss of the models with different dropout values.



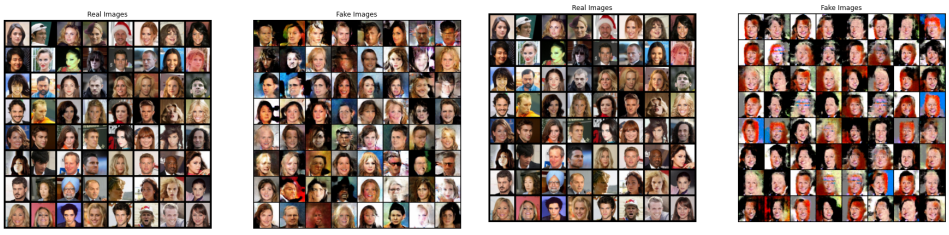
(a) dropout rate = 0.5 (b) without dropout

Figure 6: pictures generated from dropout ablation experiment



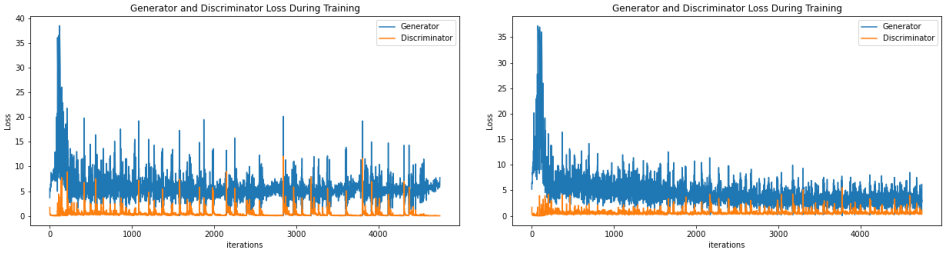
(a) dropout rate = 0.5 (b) without dropout

Figure 7: the loss of the models with dropout(0.5) and without dropout



(a) (b)

Figure 8: pictures generated by the models with different dropout values



(a) (b)

Figure 9: the loss of the models with different dropout values

6 Summary & Next Steps

In summary, we reimplemented DCGAN, tried different activation functions and conducted a series of ablation studies. We improved the DCGAN model by replacing ReLU with LeakyReLU. By conducting ablation studies, we understood the importance of batch norm and the usage of dropout in deep learning.

As for next steps, we would like to try Instance-Conditioned GAN. Generative Adversarial Networks (GANs) can generate near photo realistic images in narrow domains such as human faces. Yet, modeling complex distributions of datasets such as ImageNet and COCO-Stuff remains challenging in unconditional settings.The authors of IC-GAN take inspiration from kernel density estimation techniques and introduce a non-parametric approach to modeling distributions of complex datasets.They partition the data manifold into a mixture of overlapping neighborhoods described by a datapoint and its nearest neighbors, and introduce a model, called instance-conditioned GAN (IC-GAN), which learns the distribution around each datapoint.

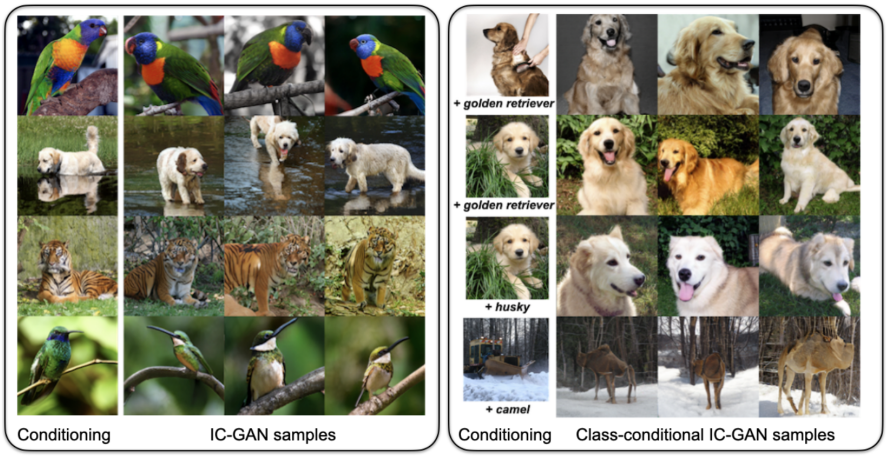


Figure 10: examples of IC-GAN

References

[1] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. DOI: 10.48550/ARXIV.1406.2661. URL: https://arxiv.org/abs/1406.2661.

[2] Alec Radford, Luke Metz, and Soumith Chintala. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2015. DOI: 10.48550/ARXIV.1511.06434. URL: https://arxiv.org/abs/1511.06434.