CSCE 479/879 Homework 3: Generative Models

Izzat Adly, Ryan Bockmon, Quan Nguyen

April 3, 2021

Abstract

We were tasked with building a generative model on the CelebA data set. We decided to use a generative adversarial network for this homework. In the end our generative model got better over time but was never able to generate images that could fool the discrimination model. We noted that there is still rooms for improvements and plan to look into using different model structures that would be better suited for image learning.

1 Introduction

We were tasked with building a generative model for the CelebA data set[1]. We had to build a generative architecture using either a variational autoencoder (VAE), or a generative adversarial network (GAN) and decided to use a GAN for this homework. We looked at the loss functions of two different models. The first was the loss of the Generator model and the second was the loss of both the real and fake Discriminator model. In the end our generative model got better over time but was never able to generate images that could fool the discrimination model.

2 Problem Description

We were tasked with building a generative model on the CelebA dataset [1]. The data set consisted of 202,599 number of face images of various celebrities with 10,177 unique identities, but names of identities are not given, 40 binary attribute annotations per image and 5 landmark locations. We had to build a generative architecture using either a variational autoencoder (VAE), or a generative adversarial network (GAN). We had to evaluate our model using an appropriate loss function for its model type such as ELBO for variational autoencoders and log probability for autoregressive models and normalizing flows. The motivation for this problem is that we are trying to identify the effectiveness of a specific Neural Network design in trying to generate fake images and in trying discriminate real from fake images.

3 Report Approach

For the homework, we decided to implement a GAN. GANs are an unsupervised learning task in machine learning that involves automatically discovering and learning the regularities or patterns in input data in such a way that the model can be used to generate or output new examples that plausibly could have been drawn from the original dataset [2]. There are two overall goals for a GAN. The first is to create generated images that are as real as possible based on a set of images and the second is to create a model that can detect if images are real or fake.

We created one GAN model with a single generator model and a single discriminator model. The generator model is used to generate new examples of images based on the CelebA data set, and the discriminator model tries to classify examples as either real or generated. The benchmarks for performance we used to valuate our models was using loss a function on Generator and both the Discriminator. This is was used to test how well the generator model generated images and how well the discriminator model was able to figure out which image was real or fake.

The combined node units of the each model was 400 for a total of 800 nodes. Both the generator model and the discriminator model used the hexagon shape architecture that we found to be have the best performance in homework 1 (Figure 1). Instead of using 175, 250 and 175 nodes we used 100, 200, 100 for our generator and discriminator models to see how they will perform in both generating new images and in discriminating fake from real images.

We expect to see major loss in the generative model early on as it is trying to learn complicated features of an image and slowly learn over each epoch. We expect to have very little loss early on for the discrimination model, because the generative model will not be creating the best fake images that will be easy to detect. We do not know if the discrimination model will do worse as the generative gets better. On one hand the generative model should be creating better images that are harder to tell the difference from, but on the other hand the discrimination model is also learning as time goes on.

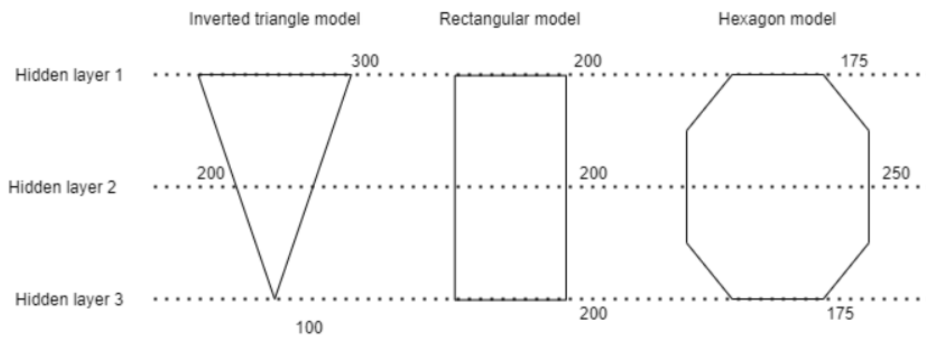

Figure 1: The three different model designs we used in Homework 1.

4 Experimental Setup

4.1 General Setup

- The data was downloaded from the CelebA data set[1].
- We ran the model for 10 epochs with 200 iterations for each epoch.
- Batch size is 512.
- At each iteration, the model randomly selected half of the batch size data (256) as inputs.
- Loss function used was "binary crossentropy".
- Optimizer was Adam with a learning rate of 0.0002.

4.2 General structure for both models

- Dense layer with 100 features
- Leaky ReLu layer with an alpha of 0.2
- Dropout layer with rate = 0.3
- Dense layer with 200 features
- Leaky ReLu layer with an alpha of 0.2
- Dropout layer with rate = 0.3
- Dense layer with 100 features
- Leaky ReLu layer with an alpha of 0.2
- Dropout layer with rate = 0.3

- (Generator) Output layer with input shape*number of channels features and activation was "sigmoid", because it had to output a new image the same size as the other images.
- (Discriminator) Output layer with 1 feature and activation was "sigmoid".

5 Experimental Results

Results from the model running was as followed

- The model took about 6 hours to run 9 epochs.
- The final loss value of discriminator was 0.0620.
- The final accuracy of discriminator was 98.63%. The accuracy fluctuated between 98% and 100%.
- The final loss value of generator was 12.1702.

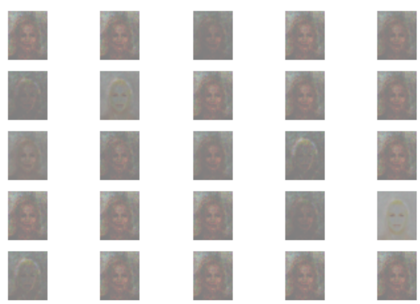

Figure 2: Generator images produced

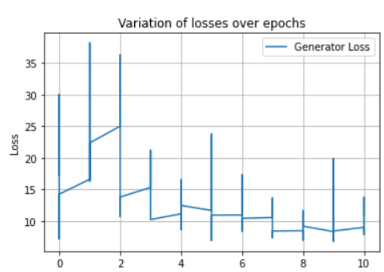

Figure 3: Generator Loss

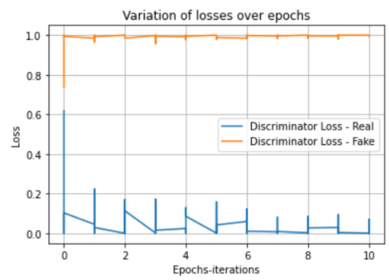

Figure 4: Discriminator Loss (Blue) and Accuracy (Orange)

6 Discussion

Looking at the images produced by the generator at figure 2, we could see that the generator was successful at producing human-like characteristics. For example, human in each images has eyes, ears, chins, and hairs. However, there were some important missing details. The most prominent one that we found was that the hair of a female (4^{th} row 5^{th} column was unnatural in that it was too thin and seemed to be part of her face). There were a lack facial expressions as all images seemed to be smiling.

From figure 3, we saw a downward trend in the generator loss values over time. At the same time, based on figure 4, the discriminator loss values for real images was close to 0 and the experimental results also showed that the accuracy values for images was between 98% and 100%; which indicated that the discriminator was good at detecting both real and fake images. Therefore, we concluded that our generator model exhibited some good results, but the generator model was not good enough to fool the discriminator.

As for learning purposes, we sat down and discussed some potentials causes and possible solutions:

1. Because we encountered some problems while loading dataset images for inputs, we devised some workarounds to circumvent the problem. However, such method may impact the overall performance as the inputs were selected randomly. Hence, the same input might get used multiple times, which slowed down learning.

2. The generator may be too simple to produce deceptive images. Because we aimed to observe how GAN model learnt to recognize real vs fake images, we picked a simpler model of only Dense layer in hexagon shape so that we could observe how the model weights changed overtime. This might impact the overall performance of the generator model as Dense layers with only hundreds nodes were not well-suited for complex images. For future experiment, we would use Convolutional layer instead as Convolutional

layer was built for image processing and learning.

3. The number of epochs could be increased to allow the model more time for training. Because we had not seen any signs of overfitting, it was possible that the generator model could continue to learn. However, due to time constraint of each CRANE's highest GPU requests, we could not let the generator model run forever. Specifically, it was relatively difficult to request the highest end GPU, which was the only GPU that we knew of that had enough RAM to process a high load of inputs and run the models. For future work, we could create model checkpoints after each small interval of training so that we could resume training at a later time.

7 Conclusions

We believed that our results were decent. Despite the limitation of Dense layer and a small number of nodes, the generator was still able to produce human-like images. While it was not good enough to overcome the discriminator as the discriminator did not have much difficulty distinguishing between fake and real images, it still showed that our model system was going in the right direction. We believed that some changes to how the model was built could significantly improve the performance of the generator.

As indicated in the discussion section, there were still rooms for improvements. We planned to use a different model structure that would be better suited for image learning. Convolutional-layer-based model came to mind as it was well suited for cifar-100 dataset (HW2). We believed that Convolutional layer would help the generator model better represented the images and enhanced learning.

Table 1: Contributions by team member for this assignment.

Team Member	**Contribution**
Izzat Adly	Wrote the original working code; Help write and proofread the report.
Ryan Bockmon	Debugged the code, and; Wrote the Introduction, Problem Description, Report Approach and Experimental Setup
Quan Nguyen	Debugged and cleaned the code; Wrote Experimental Results, Discussion, and Conclusions;

References

[1] Celebfaces attributes (celeba) dataset. accessed: 2021-04-2. URL: `https://www.kaggle.com/jessicali9530/celeba-dataset`.

[2] A gentle introduction to generative adversarial networks (gans). accessed: 2021-04-2. URL: `https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/`.