Generative Adversarial Network(GAN) Project: Face Generation

Faruk Geles

January 2020

Abstract

This project of the Udacity Deep Learning Nanodegree is about using Generative Adversarial Networks(GANs), to generate face images using a dataset from celebrities. The Generative Adversarial Network will generate new fake faces using an dicriminator and an generator neural network, where the discriminator decides if a image is real(1) or fake(0) and the generator tries to fool the discriminator generating fake images which should be accepted as real images.

1 The Goal of the Project

In this project we build a GAN model, which will use a dataset celebrity faces for training, in order to generated a fake faces script. The algorithm has two main parts:

- Definition of the discriminator architecture
- Definition of generator architecture

1.1 Used Data Set

The data which we use for this project are already provided in the project in zipped folder "$processed_celeba_small.zip$". The first step is to unzip this folder. Let us look shortly to a banch of these images

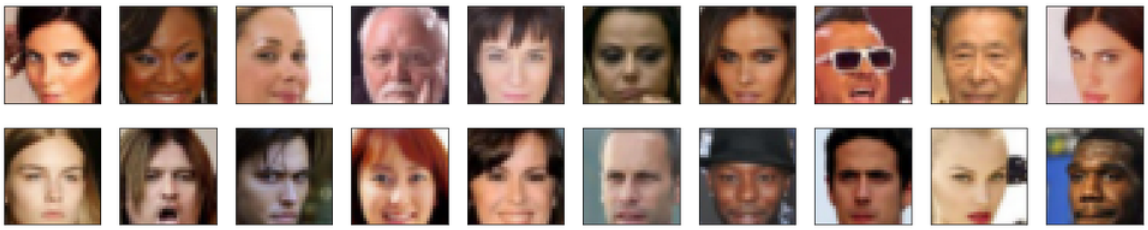

Figure 1: A set of used images in our dataset.

2 The Approach

For the TV script generation problem we use a Recurrent Neural Network(RNN)

- Prepocessing data
- Passing the data to a Recurrent Neural Network model to generate fake TV script

2.1 Data Preparation

The first thing to do on our given folder of celebrity images is:

- Resize the images if necessary making sure that they are (32x32x3)
- Converting to a tensor image type
- The data are then loaded efficiently in batches

2.2 Network Architecture

2.2.1 Discriminator Architecture

We have the following discriminator layers:

- The inputs to the discriminator are 32x32x3 tensor images
- A few convolutional layers with batch normalization
- Lastly a fully connected layer with sigmoid

2.2.2 Generator Architecture

We have the following Generator layers:

- The first layer is a fully connected layer which is reshaped into a deep and narrow layer
- Then, we use batch normalization and a leaky ReLU activation.
- Then a series of transpose convolutional layers, where you typically halv the depth and double the width and height of the previous layer.
- we'll apply batch normalization and ReLU to all but the last of these hidden layers. Where we will just apply a tanh activation.

2.3 Training

Training for our model will involve alternating between training the discriminator and the generator. Therefore we will use the functions $real_loss$ and $fake_loss$ to help you calculate the discriminator losses. We have to do the following steps:

- train the discriminator by alternating on real and fake images
- generator, which tries to trick the discriminator we should have an opposing loss function

3 Results of GAN Model for face generation

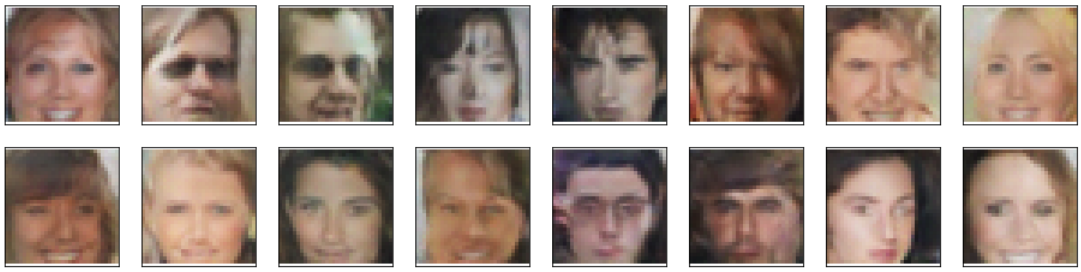

Figure 2: Some results for fake faces generated by our model.

4 Ideas for Future Work

- 1) More optimization of the parameters
- 2) More layers
- 3) Increasing the training episodes