Fall 2019 CSC 412

Programming Assignment 3: Face Generation with a GAN

Due date: Saturday, November 30th, 11:59:59PM

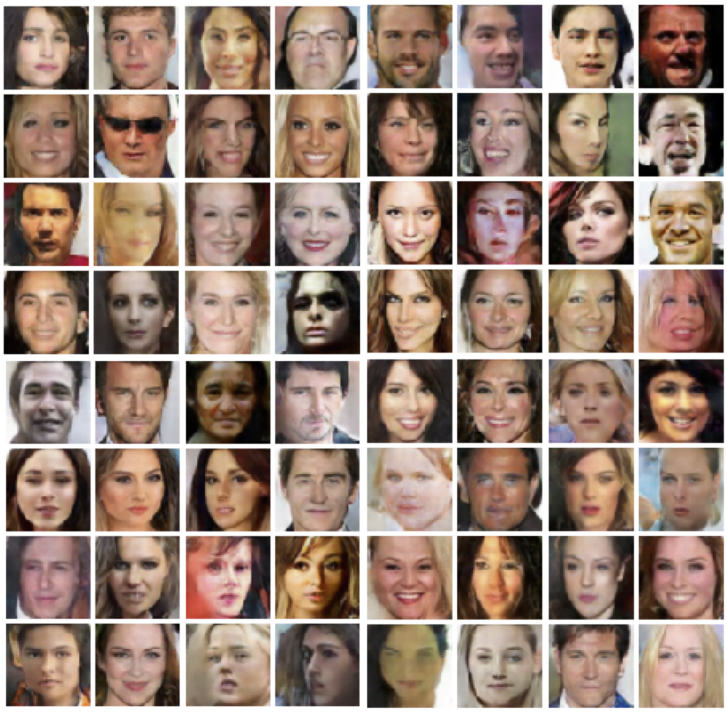
Grade:

100 marks in total, while it accounts for **20%** towards the final grade

Type:

Independent work

Credit: UIUC Svetlana Lazebnik's CS 498 and Stanford's CS231n



Sample images from a GAN trained on the Celeb A dataset

You will use a generative adversarial network to train on the CelebA Dataset and learn to generate face images.

In addition to familiarizing you with generative models, this assignment will help you gain experience with how to implement GANs in PyTorch.

This assignment was adapted from and inspired by material from the Stanford CS231n Assignments, and the PyTorch Tutorials.

Download the starting code here.

Data Setup

Once you have downloaded the zip file, go to the Assignment folder and execute the CelebA download script provided:

```
cd Assignment3/
./download_celeba.sh
```

The Celeb A data provided to you is a preprocessed version which has been filtered using a simple face detector to obtain good images for generation. The images are also all cropped and resized to the same width and height.

You will need to use a GPU for training your GAN. If you do not have access to a GPU, we recommend using Google Colaboratory to debug, but once your debugging is finished a Google Cloud machine is required as you will have to run the GAN for a few hours to train fully. For your information, Google Colaboratory provides free access to a Tesla K80 GPU card service for running short jobs.

If you are using Google Colaboratory for this assignment you will need do some additional setup steps.

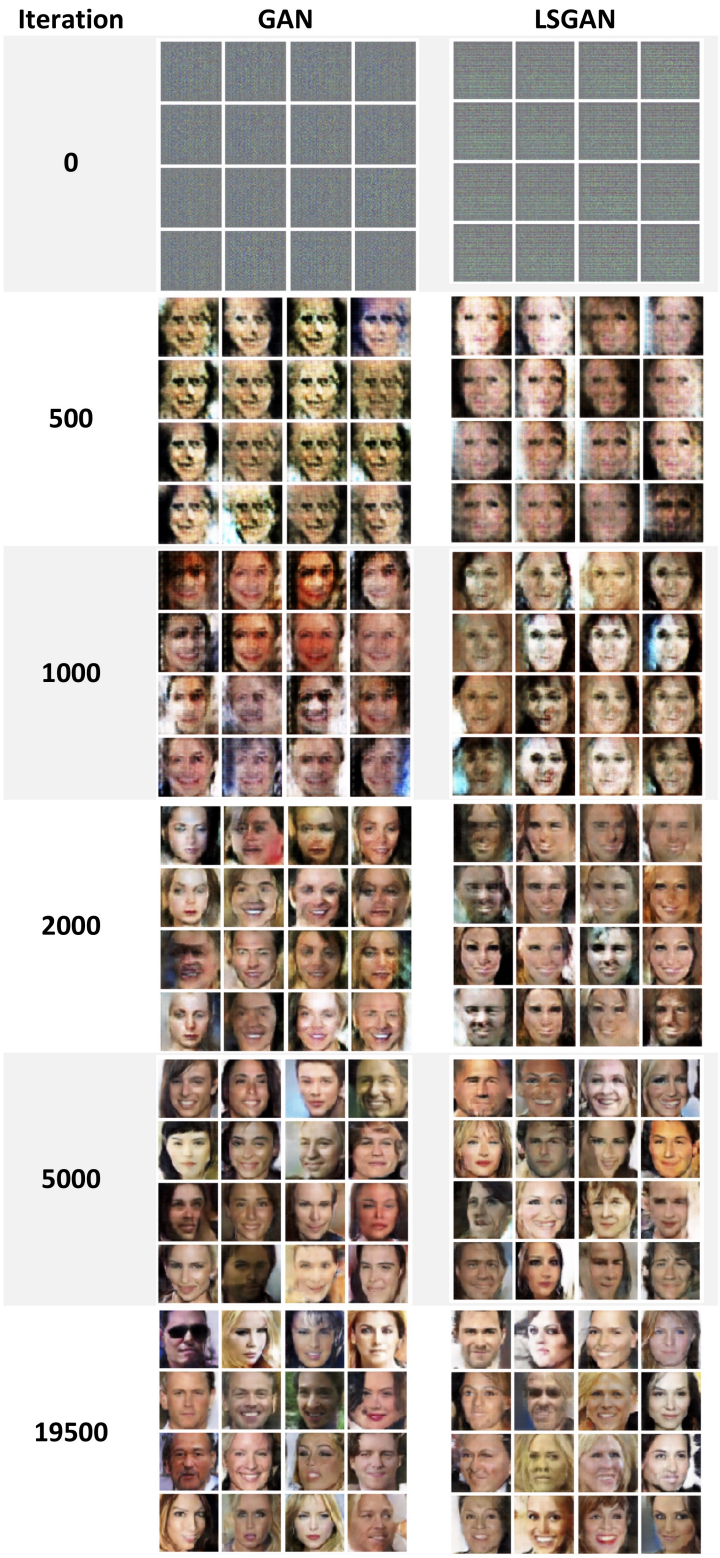
First, download the assignment zip file and follow the steps above to download starting code to your local machine. Next, you should make a folder in your Google Drive to hold all of your starting code files.

You will now need to open the Assignment 3 ipython notebook file from your Google Drive folder in Colaboratory and run a few setup commands. However, we have condensed all the important commands you need to run into an **ipython notebook** (you can disregard the section on GPU setup for this assignment).

Implementation

The top-level notebook (`PA3.ipynb`) will guide you through the steps you need to take to implement and train a GAN. You will train two different models, the original GAN and LSGAN, which has a different loss function. The generator and discriminator network architectures you will implement are roughly based on DCGAN.

We also provide with a notebook to help with debugging called `GAN_debugging.ipynb`. This notebook provides a small network you can use to train on MNIST. The small network trains very quickly so you can use it to verify that your loss functions and training code are correct.



GAN output images during training (each iteration has batch size of 128)

Extra Credit

Extra credit options for this portion of the assignment:

- Change the network architecture and hyperparameters to train on the full 128x128 resolution CelebA data in the preprocessed dataset we provide rather than the 64x64 resized samples we train on currently.
- Implement one of other the recent GAN modifications like WGAN/WGAN-GP, DRAGAN, or BEGAN.

References:

- Lecture material (lecture 12 on GANs)
- A nice blog post series including introductory posts along with posts on DCGAN and LSGAN
- Generative Adversarial Nets paper
- LSGAN paper
- DCGAN paper

Environment Setup (Local)

If you will be working on the assignment on a local machine then you will need a python environment set up with the appropriate packages. We suggest that you use Conda to manage python package dependencies (https://conda.io/docs/user-guide/getting-started.html).

IPython

The assignment is given to you in the `PA3.ipynb file.` To open the files on a local machine, ensure that ipython is installed (https://ipython.org/install.html). You may then navigate the assignment directory in terminal and start a local ipython server using the `jupyter notebook` command.

Submission Instructions

This part of the assignment is due on Compass on the due date specified above. You must upload the following files for this part.

1. All of your code (python files and ipynb file) **in a single ZIP file**. The filename should be **Student_ID_PA3_code.zip**.
2. Your ipython notebooks with output cells converted to **PDF format**. The filenames should be **Student_ID_PA3_output.pdf**.
3. A brief report in PDF format using **this template**. The filename should be **Student_ID_PA3_report.pdf**.
4. Please zip all the above documents into one package and name it as **Student_ID_PA3.zip**. Multiple attempts will be allowed but only your last submission will be graded. No email submission is accepted. We reserve the right to take off points for not following directions.

Late Policy

For every day that your assignment is late, your score gets multiplied by 0.75. The penalty gets saturated after four days, that is, you can still get up to about 32% of the original points by turning in the assignment at all. If you have a compelling reason for not being able to submit the assignment on time and would like to make a special arrangement, you must send me email at least four days before the due date (any genuine emergency situations will be handled on an individual basis).

Please refer to our course syllabus on academic integrity, and extension requests.