ENGINEERING DIVISION | NYU ABU DHABI

ENGR-UH 4560
Selected Topics in Information and Computational Systems

Deep Learning

Project 07 - GAN-Human face synthesis

Introduction

Since GANs were introduced in 2014 by Google Researcher Ian Goodfellow, the tech has been widely adopted in image generation and transfer. After some early wiry failures, GANs have made huge breakthroughs and can now produce highly convincing fake images of animals, landscapes, human faces, etc. Researchers know what GANs can do, however a lack of transparency in their inner workings means GAN improvement is still achieved mainly through trial-and-error. This allows only limited control over the synthesized images.

Dataset

You may find the dataset in the following link: CelebA
CelebFaces Attributes Dataset (CelebA) is a large-scale face attributes dataset with more than **200K** celebrity images, each with **40** attribute annotations. 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**
- **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 detection, and landmark (or facial part) localization.

Requirements

1. Program a data loader for CelebA dataset:
    a. The data loader need to be established by *torch.utils.data.DataLoader*
2. Built a generator model:
    a. The model should be established based on *torch.nn.Module* and *torch.nn.functional*
    b. The model (base model) need consist of linear layers, RELU and Tanh
    c. nn.Sequential is recommended to pack all modules in the model
    d. You can add convolution layers and other modules in the base model to push performance, but improvement should be written in your report.
3. Built a discriminator model:
    a. The model should be established based on *torch.nn.Module* and *torch.nn.functional*
    b. The model (base model) need consist of Linear layers, LeakRELU and Sigmoid layer
    c. nn.Sequential is recommended to pack all modules in the model
    d. You can add convolution layers and other modules in the base model to push performance, but improvement should be written in your report.
4. Write GAN frameworks to implement the classification pipeline:

   a. Apply the dataloader you have written in the first step
   b. Create two optimizer (SGD/Adam) for the discriminator and the generator respectively
   c. Train the discriminator and the generator alternately
   d. Define a Binary Cross Entropy loss for your implement
   e. Print out fake image prediction score and real image prediction score of your framework.
5. Write a report to introduce the structure your implement and how it works:
   a. You are supposed to report where each modules (e.g. dataloader, CNN model, loss etc.) is in your code
   b. You are supposed to plot out the loss convergence course by Tensorboard or lera.ai.

Deliverables

A zip file containing the following:

1. a working project (source code, makefiles if needed, etc)
2. a report for the detailed description of the project
    a. explain the main aspects of your code
    b. how to run your project
    c. plots and diagrams

Before submitting your project, please make sure to test your program on the given dataset.

Notes

You may discuss the general concepts in this project with other students, but you must implement the program on your own. ***No sharing of code or report is allowed.*** *Violation of this policy can result in a grade penalty.*

Late submission is acceptable with the following penalty policy:
10 points deduction for every day after the deadline