EE452 Computer Vision (Spring 2022)

Assignment 2

Generative Adversarial Network (GAN) Application

Release date: 26 March 2022

Due date: 17 April 2022

Maximum Marks: 100 (Weight in Final grade: 8%)

Instructions:

- This assignment is individual task where only collaboration in terms of discussing and idea sharing is allowed.
- A report is to be submitted on LMS describing the different phases of the work also including what worked, what did not work, what you learned, what extra things you tried and how to proceed with this etc.

Description: The purpose of this assignment is to train a Deep Convolutional Generative Adversarial Network and understand the current state of the research in GANs

Figure 1: Example Images from the Datasets

Question 1: DCGANs and more [70]

In this question we will expand on your knowledge about Generative Adversarial Networks (GANs). Building on top of what we have already implemented, the primary task in this question is to implement a Deep Convolutional GAN (DCGAN).

You can read about GANs from online resources and understand it's working principles. To get an insight on the power of advanced GANs, check out this.

Simply speaking, a DCGAN is a type of GAN that uses a CNN as discriminator whereas for the generator it uses an architecture similar to a CNN but instead of normal convolution layers, it is composed of transposed convolutions. You can read more about DCGAN in the original DCGAN paper.

To successfully complete the question, you need to complete all the parts below. Make sure to cite all the sources used.

1. Load any one of the datasets from the list below, that you think is interesting to work, on in your notebook (only the training dataset). We would recommend you guys to downscale the images i.e., 64x64 or 32x32 and if *needed,* also convert the image to greyscale. [5]

List of Datasets:

- Flowers
- Aircrafts
- Cute Dogs 🐶

2. Implement a CNN based discriminator. Justify in no more than 2 lines why have you chosen your architecture for the discriminator. Remember to use **only Pytorch** in building your networks. [12+3]

3. Implement a generator that starts with a noise vector and generates an image from it. Also, can we change the dimension of the starting vector? What effect would it have on the result of your generator (research and write your best guess, you will be evaluated for your thought process) [12+3]

4. Write the complete training loop for your GAN. Run it for 50 epochs or less, with a batch size of your liking. You can run it for more epochs, just that it would take painfully long. Your hyperparameters should be mentioned in a way that they are immediately visible. [10]

5. Determine your generator and discriminator losses in training. Plotting them would be a nice practice [5]

6. Write a function named "generate_samples" which takes a single argument "n". This function will return "n" images generated using the trained generator. Use this function to generate and display 20 images. [5]

7. This part requires you to go back to the transfer learning idea that was introduced in

Assignment 1.

a) Can we use a pre-trained VGG-16 network for this GAN? Would the resulting network still be a DCGAN? [5]

b) Since a GAN can essentially be thought of as two different networks that run-in conjunction, a discriminator and a generator. What kind of network is VGG-16, is it generative or discriminative? Why can it be not used for the other network? [5]

c) Change your GAN from the previous parts to incorporate VGG-16 in it and train it for a single time only. Does it improve your results? You can argue based on generated images and losses. [5]

8. [BONUS, max. 20 points in the assignment grade] Question 2 of this assignment introduces you to different architectures of GANs. DCGANs are almost 7 years old now, and as such aren't the current State of the art for image generation tasks. Research the current state of the artmethods on your chosen dataset and implement anyone. Compare with your DCGAN.

Question 2: Read up folks! More GANS. [30]

In this question, you will have to read up a bit on the different types of GANS that are found: Style transfer GAN, CGANS, CycleGAN, SRGANS. Only reading the abstract and introduction of these papers will equip you to solve these questions, reading them fully is up to you.

Then, go through the given situations and suggest the suitable GAN along with a short but suitable explanation:

1. A researcher gave a student the job to perform some preprocessing techniques on an image dataset. The student, while playing around with that data, applies a sufficiently large median blurring kernel to the images, but deletes the original files. These new images were padded correctly as to retain the original size of the images. Which GAN is best suited to revive the original HD images and why? [10]

2. Interns at a computer vision company are tasked with adding different skin textures to images of pandas and regular bears. If presented with a picture of a regular bear, the skin should be changed to match that of a panda. They do not have paired images of pandas and bears that they can use as direct mappings of each other. They have a set of images of pandas, and a different set of images of bears. Which GAN is best suited to achieve this image-to-image translation task and why? [10]

3. A daughter wants to give her Van Gogh-fanatic mother a present. So, she decides to turna set of family photos to a set that looks as if it's been painted by Van Gogh. Which GANis best suited to achieve this task and how would she train her model (using what datasets, etc.)? [10]

Submission Guidelines:

The entire assignment is to be done as a python notebook. Once you are done, you should upload the notebook to LMS. Create sections within the notebook in order to answer each question.

It is preferred that you do this assignment on google colab: https://colab.research.google.com/. Colab allows you to utilize its GPU, which can be accessed as follows: Runtime => Change runtime type => Change Hardware accelerator to GPU. Once you have completed your assignment on colab, you can download it as a notebook as follows: File => Download.ipynb.