

PRACTICE II:

Deep Generative Adversarial Nets

Due date: 14th January 2022

A. GOALS

This practice session pursues the following objectives:

- To understand and create your own Generative Adversarial Neural Nets
- To evolve them to create Conditional Adversarial Neural Nets

B. STUDENT WORK

This is a workgroup practice, where each group must **submit in campus**:

- A PDF document, with all the answers and explanations of your work.
- The code or notebooks derived of your work.

The final submission will be a compressed file (.zip) with the following name:
ML3_2020A1_PracticeII_X.zip; where X is your group letter.

Notes: the notebooks cited in this document are available in Campus: Additional Documentation / Practice II

1. Generating MNIST data

We will start by following the notebook. 01_dcgan.ipynb, where our goal will be to understand how is working a dcgan.

EVALUATION QUESTIONS AND CODING EXERCISES

There is no TODOs in the code. This part is for understanding and explaining the notebook.

[1 point] TODO 1. Explain the high-level idea of Generative Adversarial Nets.

[1 point] TODO 2. Both the generator and discriminator are convolutional neural nets. Which are the inputs and expected outputs of both of them before and after training?

[1 point] TODO 3. The core functions are `train` and its subfunction `train_step`. Explain step by step what they are doing.

2. Generate CIFAR10-images

Let's move to another dataset. CIFAR-10

<https://www.cs.toronto.edu/~kriz/cifar.html>

Following the notebook 02_dcgan_template.ipynb, create your own dcgan for generating new cifar-10 images

EVALUATION QUESTIONS AND CODING EXERCISES

[2.5 point] TODO 1. Complete the code for the Generator model.

```
def make_generator_model()
```

Note that now you have color images, so the output should be 32, 32, 3!

[2,5 points] TODO 2. Complete the code for the Discriminator model.

```
def make_generator_model() :
```

The final gif created after 100 epochs using a seed of 49 random samples should be similar to the attached 02_dcgan_cifar.gif

Every epoch runs in about 30/40s; so, it is OK if you use just 20/30 epochs, as far as the gif demonstrate the correct evolution – in every iteration you see more defined images.

3. CONDITIONAL GENERATIVE ADVERSARIAL NETWORKS

[1 point] Explain what a Conditional Generative Adversarial network is.

[1 point] Create your own Conditional Generative Adversarial Network to generate conditioned samples in the Fashion MNIST dataset.

You can follow the notebook `03_cdcgan_fashion_mnist_template.ipynb`.

This blog might be of a great help here:

<https://machinelearningmastery.com/how-to-develop-a-conditional-generative-adversarial-network-from-scratch/>

Note that in this blog the discriminator uses a sigmoid as the final activation function. Do not use this if you are working with `cross_entropy` with `logits=true`. If you do, you will apply sigmoid twice... and it will never converge.

The final gif created after 100 epochs using a seed of 100 random samples with labels from 0 to 9 should be similar to the one attached `02_cdcgan_fashion_mnist.gif`. Note that every column represents the same class.