Assignment (1)

24789 - Special Topics: Intermediate Deep Learning (Spring 2023)

Out Date: 2023/3/14 (Tuesday)
Due Date: 2023/3/22 (Wednesday) @ 11:59 pm EST

All exercises should be submitted to Gradescope. There are 2 assignment sections to submit in Gradescope. The first is **Homework 1**, where you submit a pdf file containing your theory answers and requested plots. The second is **Homework 1 Programming**, where you submit your codes, saved model as well as a brief report for the programming exercise. For **HW3-programming**, you should submit a zip file as the following structure:

Submission file structure

```
/andrewID-HW1
        /p1
            *.py (all the Python script files)
            p1_vae_model.pth
            p1_gan_model.pth
            p1_report.pdf
```

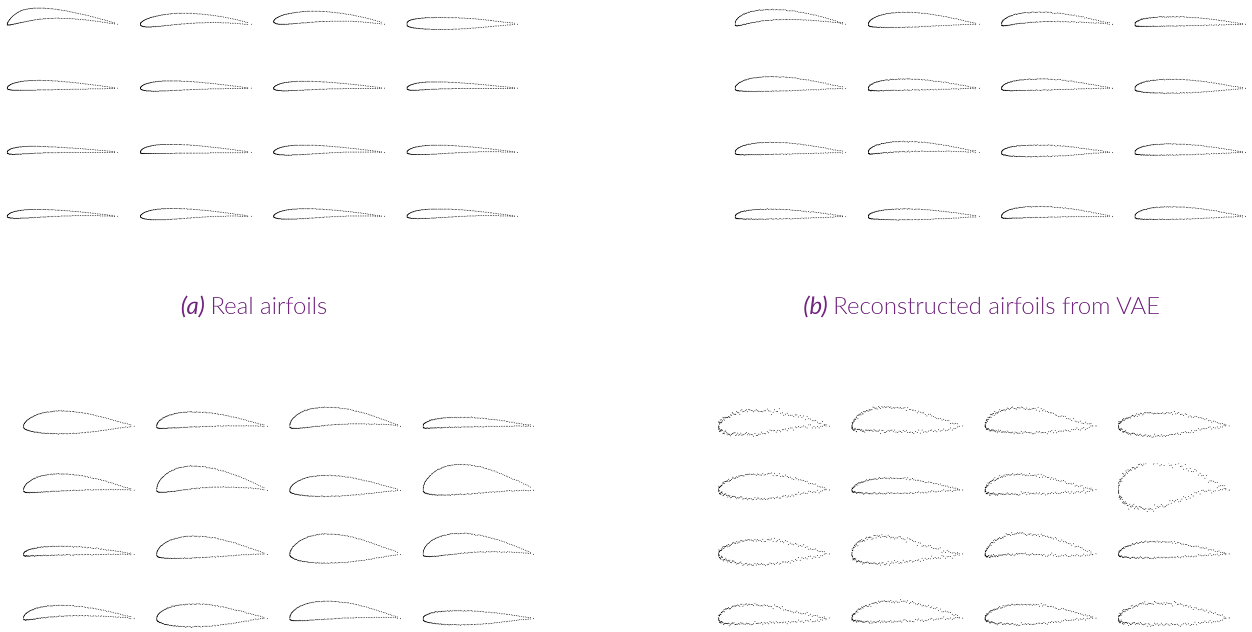
You can refer to Python3 tutorial, Numpy documentation and PyTorch documentation while working on this assignment. Any deviations from the submission structure shown below would attract penalty to the assignment score. Please use Piazza for any questions on the assignment.

Programming Exercises (50 points)

PROBLEM 1

Airfoil Generation via VAE & GAN (50 points)

UIUC Airfoil Coordinates Database includes coordinates for nearly 1,600 airfoils. Since number of points in each sample differ, we first process all airfoils to have 200 points and share the same x coordinates via spline interpolation. Also, all y coordinates are rescaled to [-1,1]. The processed airfoils are shown in Fig. 1a. Therefore, only y coordinates of each airfoil is used to train and test generative models.



(a) Real airfoils

(b) Reconstructed airfoils from VAE

(c) Synthesized airfoils from VAE

(d) Synthesized airfoils from GAN

Fig. 1: Visualization for Coding Problem 2

In this problem, you are asked to build, train and test a VAE model as well as a GAN model via PyTorch. You are encouraged to implement from the starter code. However, you are not strictly restricted to that as long as you submit all your code and complete report. Specifically, in vae.py file:

- Implement Encoder() class which takes y coordinates of airfoils as input and encodes that into a low-dimensional vector.
- Implement Decoder() class which takes the encoding vector as input and output the y coordinates of airfoils.
- Implement VAE() class which contains the encoder and decoder.

Also, you need to modify train_vae.py, which is used to train and test your VAE model:

- Define you loss function and calculate loss at each training iteration
- Tune hyperparameters both in the model and optimizer to improve the performance

Similarly, in gan.py file:

- Implement Discriminator() class which takes y coordinates of airfoils as input and predict whether airfoils are real or fake.

- Implement Generator() class which takes the normal distributed noise as input and synthesizes the y coordinates of airfoils.

Also, you need to modify train_gan.py, which is used to train and test your GAN model:

- Define you loss function and calculate loss at each training iteration
- Tune hyperparameters both in the model and optimizer to improve the performance

Also, you can write helper functions in utils.py, like generate random noises and generate labels for discriminator in GAN. Besides, since GAN loss is not directly implemented in PyTorch, you may also want to customize a GAN loss function or class.

In your report, you should include:

- Structure of your VAE and GAN (eg. number of layers, number of neurons, etc.)
- Hyper-parameters (eg. learning rate, optimizer, learning rate decay, etc.)
- For VAE, visualization of reconstructed airfoils and corresponding real airfoils as shown in Fig. 1b and Fig. 1a. Also, visualization of synthesized airfoils from random noise should be included too, as shown in Fig. 1c (starter code already has this visualization script)
- Visualization of training loss vs. epoch (both VAE and GAN)
- For GAN, visualization of synthesized airfoils from random noise as shown in Fig. 1d (starter code already has this visualization script)
- Compare the synthesized airfoils from VAE and GAN, describe your observation and give a brief explanation.

You can refer to this Github repo for implementation of VAE: `https://github.com/pytorch/examples/tree/master/vae`, and `https://github.com/eriklindernoren/PyTorch-GAN/blob/master/implementations/gan/gan.py` for implementation of GAN.