Exercise M5 VR: One-dimensional GAN

Master in Computer Vision, Course 2021-2022

Diego Porres

dporres@cvc.uab.es

Computer Vision Center, March 2022

Background

- Nowadays, GANs are able to synthesize realistic *high resolution* images
- As seen in the lectures, GANs have taken the area of unsupervised image generation by storm
- As an example, in Figure 1 we generate, in an unsupervised manner, urban scenes (moving away from faces)

Figure 1: Using the Audi A2D2 dataset, we train a StyleGAN2 to generate RGB images of size 1024^2. The grid illustrates the style transfer properties of this GAN.

Goals

- Generate simple one-dimensional data in order to focus on how the *training loop* of a GAN is implemented
- Continue enhancing your `PyTorch` skills
- Discuss some of the pitfalls and difficulties one usually encounters whilst training a GAN (mostly in the Appendices of the attached Colab notebook)

Resources

- The prepared notebook can be found here
- Please download the notebook to your local machine or make a copy in your Drive in order to continue working in Colab
- Do the assigned exercises in groups of up to 4 students
- The notebook itself is meant to be self-contained, so you will find many relevant articles and links in its main body
  - If you find any errata, please let me know as soon as possible!
  - If you have any questions, don't hesitate to email me!

Exercise 1 - Training our 1D GAN

- As we can see in Image 2, our baseline model will fail to converge to the real data distribution

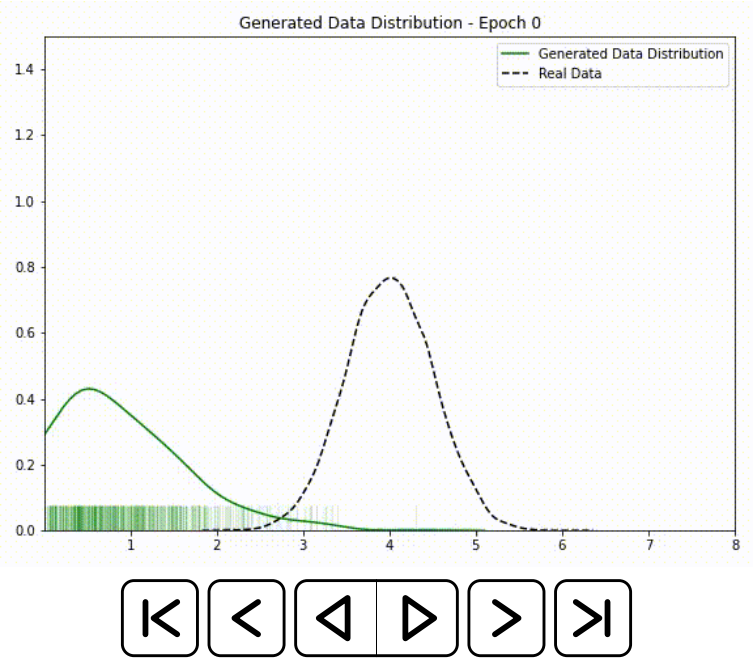


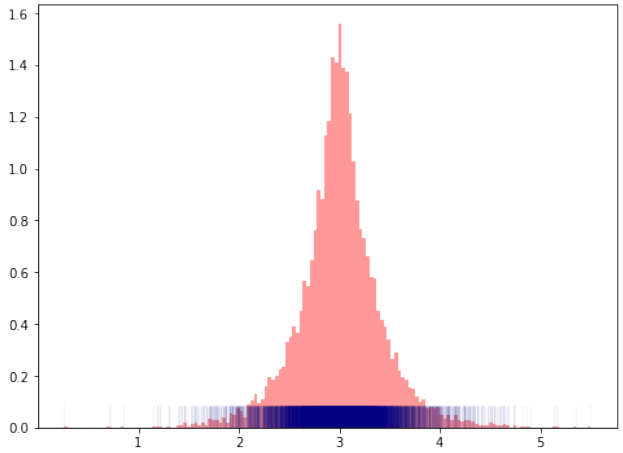
Figure 2: GIF of our 1D GAN training to imitate a dataset with Gaussian Distribution. The controls for this GIF only work on Adobe Acrobat.

Exercise 1 - Training our 1D GAN

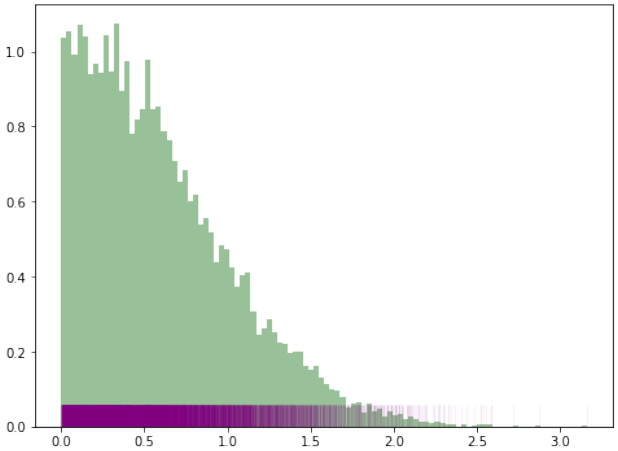
- Exercise 1 tasks you to find the correct set of parameters, hyperparameters, NN architectures, etc., in order to correctly mimic the real data distribution at the end of training
- Try to minimize the number of epochs needed to train the GAN, but there won't be a penalty if you use however many you need
- **Marks:** 0.50 points

Exercise 2 - Training with a Harder Distribution

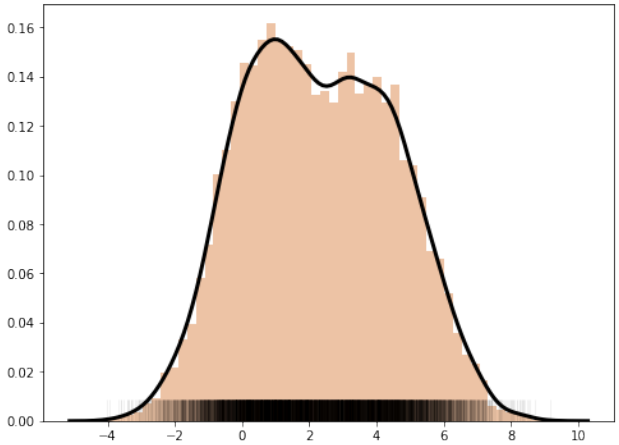
- Do the same as in Exercise 1, but now your data will follow a *harder* distribution
- You are free to choose one of the three provided distributions or mix them up as you please
  - The code to generate each distribution is provided
- Note that the last distribution does not really exist, but is a machination of my mind in these times of quarantine
- **Marks:** 0.50 points



Laplace distribution



Half-normal distribution



Petit Prince distribution

Instructions

- For Exercise 1, report any conclusions you arrive to during your trials.
  - For example, what happens if you train the baseline for longer, increase/decrease the latent dimension,increase/decrease the number of neurons per network, etc.?
  - A clear story of how you select each parameter should be shown
  - A list of suggested parameters/hyperparameters to tune is provided in the notebook
- For Exercise 2, report only the final set of parameters, hyperparameters, network architecture, etc. that you used, plus the training GIF/video as in Figure 2.
  - A concise explanation of your experiments is also expected here

Instructions

- Upload to the Virtual Campus both a presentation (in PDF format) containing your final results and experiments, as well as the notebook with your selected set of parameters and hyperparameters that yielded a good result for both exercises
  - In your presentation, you can show your results in any way you so desire (plots, GIF, video...)
  - You can either add a link to your final notebook in your presentation, or download and attach the `.ipynb` file
- Don't forget to add your team member's names and emails!
- **Deadline:** April 22nd at midnight