097200 - Deep learning: Generative Learning

Submission date: 26/01/2020

1. Exercise

In the following exercise, you will deal with a generative learning problem, precisely, VAE and GAN. You should write your training code and meet the following constraints.

2. Generative Network

In this exercise, you will create a generative model with both **discrete and continuous** latent spaces and investigate their encodings.

- choose VAE **or** GAN
- **implement and train your model:** The decoder/generator should get as input (z_discrete, z_continuous) and produce an image. There are many ways to implement this model. For example, whether the encoders' weights are shared or not, or how to combine the two latent spaces (similar to the VQA problem). For convolutions that upscale the input's spatial size (for the decoder/generator), use `nn.ConvTranspose2d`.
- **Outputs visualization:** Generate images from your model and visualize its latent spaces. You can compare different architectures for this purpose (e.g., low/high dimension of the latent spaces, expressive/inexpressive encoder, etc.). We encourage you to get creative with how you combine the latent spaces and visualize them.
- **Dataset:** You will use the CelebA dataset (one of the built-in datasets of PyTorch), which is a collection of images of faces annotated with 40 binary attributes (male/female, smiling/not smiling, etc.). You can use the attributes (labels) for training and visualizations. However, be sure to clarify how and why they were used. Images can be resized for efficiency but not smaller than 64x64.

3. Submission instructions

Submission will be in pairs (course partners) and will contain a short pdf report containing:

1. Model architecture description and illustration, training procedure (hyperparameters, optimization details, etc.).
2. Training convergence plots as a function of training time:
   GAN: discriminator and generator losses, VAE: reconstruction loss, and KL divergence.
3. Summary of your attempts and conclusions. Your conclusions and explanations should be based on the actual results you received during your attempts. Include 1-2 pages of visualizations (the images your model produces and visualizations of the latent space).

Besides, you should supply:

1. Code (python file) able to reproduce your results
2. The trained network with trained weights (.pkl file). If the model size is less than 500MB, you

should submit it on Moodle. Otherwise, upload it to your Google-Drive.

3. A function called *"reproduce_hw3()"*. This function should be able to reproduce the results that you reported.

Moodle submission:

You should submit a Zip (not Rar!) file containing:

- Code – as many files as you need (one of them should be "main.py," which will include the running process)
- One pdf file (including your names and ids)
- The .pkl file (If the file is too big for the Moodle, upload it to your Google-Drive and copy the link to your pdf report)

4. Grading policy

1. Successful submission – 50 points.
2. Report - 50 points.
3. Bonus – Two teams will earn 20 extra points (VAE/GAN team) for the most creative ideas and insightful visualizations (judged by the class staff). The maximum grade for this assignment is 120.

Good luck!