Deep learning – HW2

Submission date: 12/01/2022

1. **VAE** - In this exercise, you will implement three VAE models with latent spaces that are: (1) continuous, (2) discrete (3) both
   - **Model:** There are many ways to implement this model. For example, you may consider whether the encoders' weights are shared or not, or how to combine the two latent spaces. For convolutions that upscale the input's spatial size (for the decoder/generator), you may use `nn.ConvTranspose2d`. Note, there is no restriction to what layers you may use.
   - **Dataset:** You will use the MNIST dataset. Differently, this time you will randomly color each sample with a different color. You can implement this using a Dataset class that is wrapping the MNIST dataset (this is a suggestion; you can do it however you like if it is correct).
   - **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 latent spaces, expressive/inexpressive encoder, etc.). Demonstrate how to control the color of the output sample using the latent space. We encourage you to get creative with how you combine the latent spaces and visualize them.

   In your report, you should discuss the differences between the models and results. Moreover, how did you control the color and digit of the generated samples? What worked better and how did you evaluate it?
2. **GAN** - In this exercise, you will implement cycle GAN
   - **Model:** In this exercise, you will use CNN.
   - **Dataset:** You will use the MNIST dataset. Different than the previous question, here you should color each digit with a predetermined color. E.g., one with red, two with yellow, etc.
   - **Outputs visualization:** Generate images from your model and visualize its latent spaces. Now, demonstrate how a cycle GAN generates a digit with a color that belongs to another digit. For example, if we use in the training set red for the digit one and the color yellow for two. We would like to generate a new 'one' sample with yellow.

   Discuss and demonstrate your results in the report.

Submission instructions:

Submission **must be individual** and will contain a short (two pages) pdf report containing:

- Model architecture description, training procedure (hyperparameters, optimization details, etc.).
- A short summary of your attempts and conclusions.

In addition, you should also supply:

- Code (python file) able to reproduce your results - we might test it on different variants on these datasets.
- Trained networks with trained weights (.pkl file).
  [the weights tensors can be saved with *torch.save({'w1':w1, 'w2':w2 }, 'path_to_w.pkl')* and load with *torch.load('path_to_w.pkl')*]

Moodle submission:

You should submit a Zip file containing:

- Python files for each practical question:
  - Training procedure, file name: *hw1_id_train.py*
  - Evaluation (generation) procedure, file name: *hw1_id_eval.py*
- 1 pdf file with
  - Your full name and ID
  - Typed answers for the theoretical part
  - A summary of the practical part
- Pickle files (If the file is too big for the Moodle, upload it to your Google-Drive and copy the link to your pdf report)

Good Luck!