HW3 – Machine Learning 2 – 097209

Rademacher Complexity & Generative Models

Submission due 8/4/2024

Question 1 – Rademacher Complexity (35 pt)

For a sample set $S = \{(x_1, y_1), \ldots, (x_m, y_m)\} = \{z_1, \ldots, z_m\}$ drawn from distribution D, with labels $yi \in \{-1, +1\}$, denote an hypothesis $h : X \to \{-1, +1\}$. We define $F = \ell \circ H = \{z \to \ell(h, z) : h \in H\}$, the set of all compositions of loss ℓ with hypotheses from the hypothesis class H.

Recall the Rademacher complexity definition:

$$R(F \circ S) = E_\sigma[\sup_{f \in F} \frac{1}{m} \sum_{i=1}^{m} \sigma_i f(z_i)],$$

and the average Rademacher complexity over sample sets of size m,

$$E_{S \sim D^m}\,[R(F \circ S)].$$

1. (10 points) What are the minimum and maximum values of the Rademacher complexity for a $\{-1, +1\}$ loss ℓ, meaning $\ell : X, h \to \{-1, +1\}$, and for which hypothesis class H are they obtained?
2. (15 points) For a sample set S, and a class of functions H such that each hypothesis predicts a label $h: X \to \{-1, +1,\}$ let

   $$\mathrm{L} = \{(x, y) \to \mathbf{1}\,(h(x) \neq y) : h \in H\}$$

   be the loss class of H with respect to the 0-1 loss (an indicator loss function). Prove that

   $$R(L \circ S) = \frac{1}{2}\, R(H \circ S),$$

   when $R(L \circ S)$ denotes the Rademacher complexity of sample S with respect to the loss set L

   $$R(L \circ S) = E_\sigma[\sup_{h \in H} \frac{1}{m} \sum_{i=1}^{m} \sigma_i l(z_i, h)]\,,$$

   , and $R(H \circ S)$ denotes the Rademacher complexity of sample S with respect to the hypothesis class H

   $$R(H \circ S) = E_\sigma[\sup_{h \in H} \frac{1}{m} \sum_{i=1}^{m} \sigma_i h(x_i)]\,.$$

   hint: you can use the identity we saw in the tutorial regarding 0-1 loss.
3. (10 points) Given a function set G, define the function class

   $$\hat{F} = \{f + g : f \in F, g \in G\},$$

   as the function class of the additions between functions of set F and G. Prove that

   $$R(\hat{F} \circ S) = R(F \circ S) + R(G \circ S).$$

Question 2 - Generative models (65 pt)

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.
In this exercise, you will create a generative model:

- choose VAE or GAN
- implement and train your model: The decoder/generator should get as input a vector z from the latent space and produce an image. For convolutions that upscale the input's spatial size (for the decoder/generator), use nn.ConvTranspose2d .
- Output 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, etc.).
- 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.). Images can be resized for efficiency but not smaller than 64x64.

How to download and use the dataset:

```
import torchvision.datasets as dsets
from torch.utils.data import DataLoader

batch_size = ...  # your batch size here
path_to_data_root = ...  # your path to the data root here
your_transforms = ...  # your transforms here

train_dataset = dsets.CelebA(root=path_to_data_root,
                             split='train',
                             transform=your_transforms,
                             download=True)

train_loader = DataLoader(dataset=train_dataset,
                          batch_size=batch_size, shuffle=True)
```

You should provide:

- Code (python file) able to reproduce your results.
- 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.
- A function called "reproduce_hw3()". This function should be able to reproduce the results that you reported.

Discussion:

Discuss your results. You should provide the following:

- Model architecture description and illustration, training procedure (hyperparameters, optimization details, etc.).
- Training convergence plots as a function of training time: GAN: discriminator and generator losses, VAE: reconstruction loss, and KL divergence.
- 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).

Submission

- Submission in pairs unless otherwise authorized.
- Solution of Q1 and the discussion of Q2 should be typed. Hand-written submissions won't be accepted.

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 (solution of Q1 and the discussion of Q2).
- 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).
- Run 'pip freeze > requirements.txt' and attach it to your submission.