Very Deep Learning

Assigment 5

January 2019

1 Generative Modeling via Generative Adversarial Networks

1.1 Theory Problems

1. What are Generative Algorithms? Name few (atleast 2) types of Generative Algorithms and give atleast one application example for each?

2. Please write informal and intuitive description of Generative Adversarial Networks. What is the goal behind them. Name the components and briefly explain what they does and the training procedure using value function.

3. Remember from the lecture, what is the global optimal solution of a GAN? By fixing generator (G), what is the optimal discriminator (D) and vice versa?

4. From the inception of GANs, they are said to be unstable in training and inference periods. Briefly explain the challenges (atleat 3) and their respective techniques to overcome them.

5. How do we measure the performance of generative models. Brief explain 2 metrics with their advantages and disadvantages?

   Min Submission : Answer all theory questions

1.2 Practical Problems

Download and extract the zip file for assignment5. Insall the require packages.

1. Open file assignment5_1_mlp_gan.ipynb.

   - Complete the code marked with ##TODO##

   - Add regularization (i.e., dropout) layers with probability set to 0.3. Compare models performance (with and without dropout) for 80 Epochs. Plot loss and score curves of both discriminator and generator. Why do we need dropout, at all here? Submit the plotted histograms.

   - By now, we know some of the crucial challenges faced by GANs. Make the generator prone to Mode-Collapse. Submit the resulting notebook as assignment5_1_mlp_gan_mode_collapse.ipynb and visualization output for any particular mode(e.g., 5).

2. We can improve the qualitative results obtained above by replacing MLP with Deep Convolutions. Open file assignment5_2_dcgan.ipynb.

   - Complete the code marked with ##TODO##

- Add batch normalization layers and check the efficiency of the network by plotting generator and discriminator loss historgrams for 20 epochs. Submit the plotted historgrams.
  - Note: Reduce in loss not necessarily mean increase in model performance. Save generated images for every few epochs and check the model performance manually.

- Replace *normal_init* with *xavier_init*. Submit the plotted histograms.

- After training for 20 epochs, add some noise(e.g., white noise) to generated images before passing them to discriminator for evaluation as real/fake (just like Adversarial-Examples). Report couple of noise images where discriminator fails to distinguish real from fake or vice-versa.

For more GAN-related information refer here : `https://github.com/soumith/ganhacks`

Min Submission : Complete task 1 and task 2. Submit the results obtained from above sub-tasks.