CS 7150: Deep Learning — Summer-Full 2020 — Paul Hand

Week 11 — Preparation Questions For Class
Due: Monday July 20, 2020 at 12:00 PM Eastern time via Gradescope

Name: Saurabh Vaidya
Collaborators:

You may consult any and all resources in answering the questions. Your goal is to have answers that are ready to be shared with the class (or on a hypothetical job interview) as written. **Make sure to tag each question when you submit to Gradescope.**

Directions: Read the article 'Generative Adversarial Networks' and 'Wasserstein GAN'.

Question 1. *Explain how GANs work and how they are different from VAEs.*

Response:

Answer 1. *GANs is a generative modeling framework. It consists of a differentiable function called generator $\mathcal{G}(z;\theta_g)$ which takes a latent code or even a random noise as input and outputs a point in data space. The output of the generator is fed to another differentiable fucntion called discriminator $\mathcal{D}(x;\theta_d)$ which outputs a probability that x came from the data distribution and not from generator's distribution. The outcome of trained GAN is a generator which has learned a data distribution p_g such that the discriminator has equal probability outcome for each input from this distribution p_g.*

Both these functions can be neural networks which are trained simultaneously. The discriminator network maximizes the probability of assigning correct label to training sample i.e. whether it was a true input from data space or an adversarial input generated by the generator cG. At the same time, the generator tries to minimize the probability of discriminator correctly classifying its output i.e it tries to make its output resemble the true data as close as possible. Once both generator and discriminator can't improve anymore, they stop and training stops.

VAEs aim to learn a latent code distribution such a generative model can be trained from the learned latent code distribution. GANs on the other hand can take a random noise as input and train a generator along with a discriminator such that it becomes good enough to geenrator data which fools a simultaneously trained discriminator.

Unlike VAE, GANs do not attempt to maximize the data likelihood. Thus we do not get a $p(x)$ which can help us evaluate a performance of GAN.

Question 2. *What is the relationship of a GAN to game theory?*

Response:

Answer 2. *As seen in the training of a GAN, the generator and discriminator are working against each other. Both of them are trying to outperform the other. This is similar to game theory where we have two players (generator $\mathcal{G}$ and discriminator $\mathcal{D}$) trying to play against each other where each player is tryig to maximize their reward. In GAN, the generator is trying to minimize discriminator's probability of* *classifying generator's output as fake and the discriminator is trying to maximizing its probabilities of identifying true data from generator's data.*

The game between $\mathcal{G}$ and $\mathcal{D}$ is a minimax game where the game would stop when the players achieve a Nash equilibrium. In GAN, nash equilibrium is when $\mathcal{D}$ cannot distinguish between real and generator's data and $\mathcal{G}$ cannot do any better at creating new data.

Question 3. *What is mode collapse? Why does it happen?*

Response:

Answer 3. *Mode collapse is a phenomenon where our generator only produces a fixed output or a small subset of output for every random input to the generator.*

Mode collapse occurs because the discriminator is not optimally trained. Thus it fails to detect a higher number of fake inputs than real inputs. This makes the generator's job easy since a small set or a fixed output always manages to fool the discriminator and thus generator need not maximize its data likelihood anymore.

Question 4. *Explain the significance of Figure 3 in the GAN paper.*

Response:

Answer 4. *Figure 3 shows a plot of Wasserstein-1 ($\mathcal{W}$) distance versus number of iterations for the generator. The significant aspect of the figure is it shows a correlation between Wasserstein-1 distance and the quality of output produced by the generator. Thus we can get a sense of the performance of our GAN by how the convergence of its $\mathcal{W}$ values.*

Question 5. *What is the difference between the training algorithm for a GAN and a WGAN? .*

Response:

Answer 5. *Both the GAN and WGAN involve training a generator and discriminator(called critic in WGAN) simultaneously. However there are two main difference's in the training algorithm.*

- *In GAN, the loss function is the Jason-Shannon divergence given by $\nabla_d \frac{1}{m} \sum_{i=1}^{m} \left[\log \mathcal{D}(x^{(i)} + \log(1 - \mathcal{D}(\mathcal{G}(x^{(i)})) \right]$ and it basically accounts to generator being succesfull at fooling the discriminator on its outputs.*

- *In WGAN the loss function is Wassertein-1 distance between the generator distribution and the real distribution. The training tries to minimize this distance and gradient update is given by $\nabla_w \frac{1}{m} \sum_{i=1}^{m} f_w(x^{(i)}) - \sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))$*

- *The weights are clipped in WGAN to be in some range $[-c, c]$ which is essential to keep the Lipschitz constant of $f_w \leq K$. This constraint is absent in plain GAN.*