Darsh Balani, Sharan Sahu, Raghav Ramanujam

EECS 182

Project Documentation Writeup

Due 12-07-22

## Key Concepts of Wasserstein GAN and Motivation of HW design

The Wasserstein GAN paper delves into the convergence issues with GANs, and describes how the usage of a different metric, Wasserstein Distance, is beneficial for the convergence and robustness of a generator. We designed our homework to explore the concepts in the paper with a mathematical rigor suited for EECS 182 while also guiding students through implementation to gain a better grip on the model.

We start with Question 1. The paper starts by describing a rudimentary MLE approach to training a generator, and describes how as the size of the data set grows, this approach is asymptotically the same as minimizing the KL-divergence between our predicted distribution and true distribution. In our homework, we have a short exercise dealing with proving this limit and engaging in discussion regarding why this poses problems concerning low-dimensional supports. While this was not explicitly covered in the WGAN paper, it is important to create some motivation for the pitfall of Vanilla GANs to understand why WGAN is an improvement. Then, the paper introduces the GAN approach, and briefly mentioned that GAN training has issues with stability and non-convergence. Expanding on this concept, our homework gives an example of a simple function that the GAN training approach fails to converge on. Further, it also addresses when the discriminator is too good, leading to vanishing gradients and inadequate training of the generator. These problems help the student better understand the theoretical foundations of the GAN and its relation to minimax games. Further, we believe that these problems complement the paper well as they explore important intuitions for the pitfalls of GANs brushed over in the paper, namely non-convergence and vanishing gradients given a perfect discriminator, a scenario that can occur in Vanilla GANs in low-dimensional supports.

Moving on to Question 2, the paper then moves on to discuss why a smooth, continuous loss function would be useful to help with convergence. We walk the students through the Wasserstein distance and a basic calculation of it. This example helps the student develop the "distance" that is being calculated as a minimum work solution of changing one probability distribution into another one, and aids with demystifying the calculation and reasoning behind the choice and design of the metric. We then ask students to calculate different distance metrics (KL, JS, and Wasserstein) on a specific example given in the paper to demonstrate the smoothness of Wasserstein distance in comparison to KL and JS divergence. This example shows that there exist sequences of distributions that don't converge under the JS or KL but do converge under the Wasserstein distance. Therefore, this question is aimed to demonstrate the superiority of Wasserstein distance in preventing vanishing gradients and in convergence. There is also a discussion of the practical issues of using Wasserstein distance in the paper. One of the problems mentioned in the paper is that it is very difficult to calculate it practically. We briefly ask why, then go on to describe that it is very difficult to find a pair of distributions that minimizes the loss as there is an infinite number of such pairs. This motivates the Kantorovich-Rubenstein Duality mentioned in the paper, which allows us to optimize over a much smaller subset of function, name 1-Lipschitz continuous functions.

For Question 3, we explore a potential improvement on the WGAN related to the Kantorovich-Rubenstein Duality. It turns out that weight clipping is a terrible way to enforce a Lipschitz constraint because it can cause WGAN to suffer slow convergence after weight clipping (when the clipping window is too large) and vanishing gradients (when the clipping window is too small). This ends up leading to undesirable results such as vanishing gradients and slow convergence. Thus, we explore a possible modification of the loss function called the Gradient Penalty that remedies this.  We also developed some intuition on why this gradient penalty solves the same problem as WGAN but with easier optimization constraints. Not only does his question help the student develop an intuition for possible forward directions from the paper, but it also helps to solidify the practical implementation details for the WGAN and understand the complexity of GAN training. While it is out of scope for the WGAN paper and does understand that this question is fairly theoretical and rigorous as it makes use of some results from convex optimization and multivariate calculus, we included this question because it is regarded as an improvement to the WGAN model. Furthermore, it is also important to motivate this because students will be implementing it in the coding question due to better convergence and results. Overall, the analytical portion of the homework is designed to expand on crucial concepts that are quickly brushed over in the paper to help students develop an intuition for design choices in GANs and WGANs. It was also designed to complement the steps taken in the coding/implementation part of the homework so that the steps to do in that part are more understandable.

The coding part of the project involves implementing a GAN and WGAN on a dataset of 30,000 greyscale images of celebrity faces. If the coding implementation for the GAN and WGAN is correct, then the model will output generated greyscale faces after 100 epochs of training. This coding implementation primarily focuses on and tests the students on the construction of the GAN loss function. Specifically, many of the empty code statements require the students to calculate the generator or discriminator/critic loss. In the GAN coding file, the loss functions for both the generator and discriminator give notes regarding the equivalence between the minimization of cross entropy and KL divergence. The homework also hints at whether or not the minimized loss should be positive or negative due to it being a min-max function. The GAN part of the coding section does not test particular parts of the paper, but rather helps students learn about the basic workings of the GAN and how the generator and discriminator interact with each other in their min-max relationship. Due to the nature of GANs, students might run into convergence errors in the middle of training the model. This helps to demonstrate one of the downfalls of GANs since they suffer problems such as non-convergence, vanishing gradients, low-dimensional supports, weight clipping issues, etc (We also have specified that this may occur in the homework itself so they are not caught off guard). In the WGAN coding file, the students are tested on the critic loss and gradient penalty. This helps to demonstrate the difference between GANs and WGANs as the loss functions are different from one another. The gradient penalty is out-of-scope of the paper, but it is in-scope for the homework since it is not that much of an extension compared to WGAN. Most of the concepts explored in the gradient penalty are concepts and techniques found in EECS 127, and it is extremely beneficial to know that an improvement to WGAN exists and why it works.