Depth First Learning **Wasserstein GAN**

Week 3: 17 Feb, 2019

Scribed by: Sasha Naidoo & Julia Rozanova

3.1 Maxing out JSD

It was mentioned in the paper this week that as we decrease noise, $\mathbb{P}_X$ and $\mathbb{P}_{X+\epsilon}$ become more similar. However, we see that $JSD(\mathbb{P}_X||\mathbb{P}_{X+\epsilon})$ is maxed out, despite the amount of noise.

In order to understand, we first think of $\mathbb{P}_X$ lying in a low dimensional manifold. Once we add some noise ϵ, $\mathbb{P}_{X+\epsilon}$ is now no longer in this low dimensional manifold. $JSD(\mathbb{P}_X||\mathbb{P}_{X+\epsilon})$ is therefore maxed out (when ϵ is non-zero) as one distribution lies on a low dimensional manifold while the other does not. The JSD is typically upper bounded by $2log2$ so the phrase "maxed out" refers to reaching this bound.

3.2 Questions on Terminology

3.2.1 What is a manifold?

Some intuitive descriptions were discussed, with the core idea being that it is a subspace of a higher dimensional real space that locally (on a small patch around any given contained point) resembles the Euclidean plane. Examples: sphere, torus (donut) etc.

3.2.2 What is the "support" of a distribution?

The intersection of all closed sets for which the probability measure is equal to 1 (equivalently, the smallest closed set which has probability 1). There was some confusion between distributions, densities and the definition of support. It was thus resolved: in the case where the distribution *has* a density function (not the case for lower dimensional manifolds), the idea of the "support" of a function (as defined on wikipedia) is equivalent. Otherwise, we need to use Martin's definition of "support"; he also suggests avoiding saying that "the support IS" but rather "the support is contained in".

3.2.3 Integrals

Lebesgue integrals were a cause of some confusion. It was suggested that we think of them as expectations instead. For the example at the bottom of page 9 of "Towards Principled Methods for Training Adversarial Networks", it was advised that we see it as an expectation over "y sampled from $\mathbb{P}_X$". This integral cannot be written as the integrand times a probability density times dx, as this distribution does not have a density; this is the main reason we are required to work with Lebesgue integration rather than ordinary calculus.

3.3 On Min/Max and Max/Min Games

- It was established that MinMax and MaxMin games are not the same, with a helpful example from Martin in the same context in which we are currently working: he suggested that it was helpful to consider a probability distribution with two point masses, for example valued $\frac{1}{3}$ and $\frac{2}{3}$. The difference between treating the training as a MinMax game as opposed to a MaxMin game could be seen as the difference between training a discriminator to optimality and training the generator in small steps, or on the other hand training the generator to optimality and training the discriminator in small steps.
- Martin answered a question regarding Nash equilibria, clarifying that both the MinMax and the MaxMin have the same Nash equilibrium here.

3.4 Change of Variables

In the first step of the derivation on page 4 of the above-mentioned paper, it seemed that a Jacobian was "missing" in the interchange of variables from one step to the next. This was briefly discussed: the first suggestion was that it is implicitly included in the "dx" term, and secondly it was suggested that the transformation from the first line to the next was intuitive if considered in terms of expectations.

3.5 Using GANs to minimise divergences we cannot compute

It was found that in practise you cannot use the reverse KL divergence or the JSD because we need the true data distribution. When training GANs, however, we are able to approximate the JSD without knowing the true data distribution. Refer to a blog post in the optional reading for this week for more detail.

3.6 Generalisation/Overfitting of GANs

While it is possible for a GAN to learn a data distribution, it essentially is learning based on samples. In the realistic case where we do not know the true data distribution, the only real solution for evaluating GANs is to compare to a hold out set. We then have to consider the likelihood of getting images that a GAN hasn't seen as the GAN will assign 0 probability to these new images.

Additional solutions, such as Parzen windows, allow the GAN to determine how similar or non-similar the new images are to existing images to avoid returning 0 probability results. Alternatively, interpolating in the latent space could also help identify whether or not the GAN has just "memorised" the training set as it should be able to smoothly move between images. We also note that pure image quality doesn't necessarily mean that the GAN was able to learn a good distribution of the data.

Furthermore, while humans are capable of identifying whether a generated image is "good" or not, when it comes to other types of data, we do not necessarily have an independent technique/score for evaluating the GANs ability to generalise.

Despite these techniques, generalisation of GANs as well as a concrete way of determining whether you are overfitting, is still a fairly unsolved problem.

3.7 Scheduling of the Generator and Discriminator

When we have low dimensional data manifolds, you can have a discriminator that perfectly differentiates between the two distributions. If the discriminator is always correct, then this leads to vanishing gradients. This is a problem with over-training the discriminator. Conversely, if you do not train the discriminator enough, this results in a lack of clarity around what is being minimised, which can cause poor quality feedback for the generator.