Project Checkpoint

The Mathematical Engineering of Deep Learning

Rohan Hitchcock, The University of Melbourne

The paper we will explore in this project is "Wasserstein GAN" by M. Arjovsky, S. Chintala and L. Bottou[1]. This paper was chosen because it uses a theoretical concept which is interesting in its own right (the Wasserstein metric on probability distributions) to attempt to solve an practical problem (making generative adversarial networks easier to train). As such this paper is interesting from both a purely mathematical perspective and from a practical machine learning perspective. The paper explicitly describes how the theory informs its approach to the practical problem, however we believe that the practical methods of the paper can also provide tools for a reader who is interested in learning only about the theoretical concepts. This is reflected in our goals for the computation demonstration.

The computational demonstration of this project has two primary goals, both of which will be implemented in Python. The first is to produce a 'toy' Wasserstein GAN for pedagogical purposes. This program will allow a user to specify a family of generators $\{f_\theta\}_{\theta\in\Theta}$ and true parameter $\theta_0 \in \Theta$. The program will then train a discriminator for $f_\theta(Z)$ and $f_{\theta_0}(Z)$ for various values of θ (where Z is some noise distribution, and $\Theta \subseteq \mathbb{R}$). This would then be used to produce plots of the Wasserstein distance between $\mathbf{P}_\theta \sim f_\theta(Z)$ and $\mathbf{P}_0 \sim f_{\theta_0}(Z)$ (up to scaling by some positive constant, see equation 3 page 7 of the paper) as a function of θ. Compared to other readily available methods for computing Wasserstein distance[2], this approach has the advantage of not placing any restrictions on the $\{f_\theta\}_{\theta\in\Theta}$. In particular the $f_\theta(Z)$ may be multivariate, singular, and their supports are not required to be equal. As such, this program would be a useful tool for a reader interested in investigating the Wasserstein distance, even if they are not interested in generative adversarial networks. The discriminator functions learned for different values of θ may also be interesting from a pedagogical perspective.

The second goal is to implement the WGAN algorithm described in the paper. In order to fit within a relatively small computational budget these implementations will be trained on a simple dataset, with simple architectures for the generator and discriminator. Initially this will be attempted using the MNIST dataset and small, fully connected networks. This, however, may require some adjustment to ensure: (1) the problem is simple enough to fit within the computational budget, and (2) the architectures of the generator and discriminator are sufficient to capture the complexity of the solution. The WGAN approach would then be compared with the so-called 'Vanilla' GAN introduced in class.

As a whole, this project will hopefully be useful to a reader who wishes to learn about Wasserstein GANs and the theory behind them. In the first part, the key theoretical concept – the Wasserstein metric – is introduced along with tools to gain an intuitive understanding of it. This concept is then applied in the WGAN algorithm, which can be compared with other approaches.

[1] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017. arXiv: `1701.07875 [stat.ML]`.
[2] One such example: scipy.stats.wasserstein_distance