Machine Learning Club, Summer Induction Assignment, 2023

1 Abstract

The authors propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation.

2 Motivation

The motivation behind this paper stems from the desire to develop a robust framework for estimating generative models. Traditional approaches often face challenges in accurately capturing complex data distributions. In response, the authors propose a novel framework based on an adversarial process. By training two models simultaneously—a generative model (G) and a discriminative model (D)—the framework aims to improve the accuracy of capturing the underlying data distribution.

The key motivation for this approach lies in the need to overcome limitations of previous methods. Many existing techniques rely on heuristics or assumptions about the data distribution, which may not always hold in practice. Additionally, certain approaches, such as those utilizing Markov chains or unrolled approximate inference networks, can be computationally expensive and impractical for large-scale problems.

The proposed framework offers several advantages, including the ability to train the models using backpropagation, which is efficient and widely applicable in the context of neural networks.

3 The Generative Model

The generative model in the proposed adversarial modeling framework is implemented using a multilayer perceptron. The goal is to learn the generator's distribution, p_g, over the data x. To achieve this, a prior is defined on input noise variables, $p_z(z)$. The generator function, denoted as $G(z;\theta_g)$, maps the input noise z to the data space. It is represented by a differentiable multilayer perceptron with parameters θ_g. The objective is to optimize G in such a way that it generates samples that resemble the training data.

4 The Discriminative Model

In the proposed adversarial modeling framework, the discriminative model plays a crucial role in assessing the likelihood that a given input x originates from the real training data instead of the generator's distribution, p_g. The discriminative model, implemented as a multilayer perceptron $D(x;\theta_d)$, outputs a single scalar value.

During training, the objective of the discriminative model is to maximize the probability of assigning the correct label to both real training examples and generated samples from the generative model G. By maximizing the probability of correct classification, the discriminative model enhances its ability to differentiate between real and generated data.

To optimize the discriminative model, an iterative numerical approach is employed. However, completing the optimization of D in each iteration is computationally prohibitive and can lead to overfitting on finite datasets. Therefore, an alternating approach is adopted, involving k steps of optimizing D followed by one step of optimizing G. This strategy ensures that D remains close to its optimal solution, as long as G changes gradually.

5 Loss Function

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{1}$$

6 Nash Equilibrium

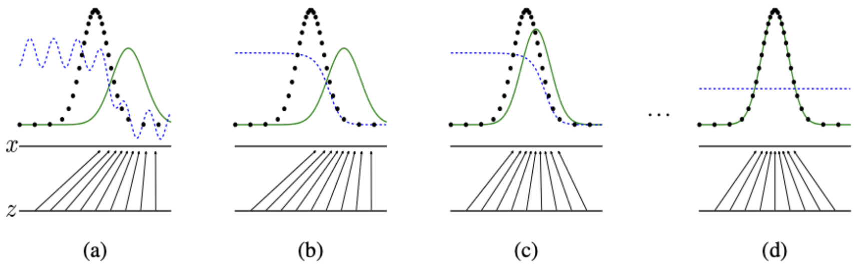

Figure 1: Illustration of the training process in generative adversarial nets (GANs). The discriminative distribution (D) is updated to differentiate between samples from the data generating distribution (p_x) and the generative distribution (p_g) represented by G. The mapping $x = G(z)$ transforms samples from a uniform distribution (z) to a non-uniform distribution (p_g). As G and D iteratively improve, they converge to a point where p_g becomes similar to $p_d ata$, and D cannot differentiate between the two distributions ($D(x) = 1/2$).

7 Algorithm

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets

for $iteration \leftarrow 1$ number of training iterations **do**
  for k steps **do**
    Sample minibatch of m noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_g(z)$
    Sample minibatch of m examples $\{x^{(1)}, \dots, x^{(m)}\}$ from data generating distribution $p_{data}(x)$
    Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D(x^{(i)}) + \log \left(1 - D(G(z^{(i)})) \right) \right] \tag{2}$$

  end for
  Sample minibatch of m noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_g(z)$
  Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log \left(1 - D(G(z^{(i)})) \right) \tag{3}$$

end for

8 Conclusion

In conclusion, the paper presents the concept of generative adversarial networks (GANs) and their training procedure using the adversarial modeling framework. The authors propose a framework where a generator model and a discriminator model are trained simultaneously in a two-player minimax game. The generator aims to generate realistic samples from a prior distribution, while the discriminator aims to distinguish between real and generated samples.

The paper highlights the effectiveness of using multilayer perceptrons as both the generator and discriminator models in the GAN framework. The training objective involves maximizing the probability of assigning correct labels to real and generated samples. The authors explain the training process, which involves alternating between optimizing the discriminator and the generator. They also provide a theoretical analysis of GANs, demonstrating that with sufficient capacity, the generator can recover the data generating distribution.