Critique - Generative Adversarial Nets

Atul Balaji
Department of Electrical Engineering
Indian Institute Of Technology, Madras
ee16b002@smail.iitm.ac.in

Abstract

In this critique, the Generative Adversarial Nets paper is systematically analyzed, exploring its core features and limitations. The paper is first summarized, emphasizing its major research contributions in deep learning. Some of the paper's shortcomings like difficulties in training and high sensitivity to hyperparameters are then analyzed. Finally, some steps are suggested to overcome its shortcomings.

1 Introduction

Among the deep learning models available, the class of models known as **discriminative** models have achieved the greatest success on both image and natural language data. However, **generative** models have not been as successful. In the Generative Adversarial Nets research paper, a new architecture is proposed, which involves a generator (a neural network that learns to produce data) and a discriminator (a neural network that learns to distinguish between real content and generated content) which are **simultaneously trained** against each other, lending it the name - Generative Adversarial network.

The generator (G) is a simple neural network that converts random vectors into meaningful data (images, word vectors). The discriminator (D) is also a simple neural network that takes the data (say image) as input and produces the probability that the sample came from the training data rather than G (between 0 to 1), 0 being certainly fake data and 1 being certainly from the training data.

2 Overview

The aim of the Generative Adversarial Nets paper is to define and implement the Generative Adversarial Networks architecture and explore its applications and performance on a range of datasets. It introduces two neural networks - the Generator and the Discriminator, which work against each other, analogous to a thief who manufactures counterfeit currency and a cop who must distinguish the fake currency from the real one, the Generator being analogous to the thief and Discriminator to the cop. This framework represents a two-player minimax game, which in time results in the generator producing convincing data, similar to the input used for training.

2.1 Discriminator and Generator

The generator network G(z;θ_g) is represented by a neural network with parameters θ_g. It takes the input noise distribution $p_z(z)$ and creates a mapping to the data space ($p_g(x)$). The discriminator D(x;θ_d) is represented by a neural network with parameters θ_d. It takes an image as input and returns the probability that it came from the training set, rather than p_g.

G and D are the players of a zero-sum minimax game, the optimum of which involves G being able to generate images very similar to the dataset and D outputting 1/2 for all inputs. This results in a Nash Equilibrium where the actions of the opponent will not change the game's outcome regardless.

2.2 Training Process

G is trained to **minimize** $\Sigma log(1 - D(G(z))$.
D is trained to **maximize** $\Sigma log(D(x) + log(1 - D(G(z))$, which is the probability of assigning the correct labels to the input image (1 for training data and 0 for samples from G).

2.2.1 Generator Training

We first create B random vectors (B is the batch size) of some particular shape. These vectors are forward propagated through the network, and new images are generated as output. These are used to train K (hyperparameter) steps for discriminator. The losses obtained are then backpropagated through G and a stochastic gradient update is performed to train G.

2.2.2 Discriminator Training

B/2 images are taken from the training set (label 1) and B/2 images from p_g (label 0). This is merged into the batch and is fed through D. Cross-entropy loss is then computed for D, which is then back-propagated through the network. The weights of D are also updated using gradient descent.

3 Merits

- Unlike other generative models, this paper proposes a methodology without Markov chains, but using only backpropagation which has been used very successfully in training deep neural networks.
- Inference is not required during learning.
- The generator learns only through the gradients backpropagated through the network. Therefore, it cannot memorize the training dataset, hence is not prone to over-fitting.

4 Shortcomings

- **Does not extend to text data** - The training algorithm described relies on backpropagation for training and thus requires a differentiable function [3] (by necessity continuous). This works in images since the pixel values can be changed by any amount. However, in **Natural Language Processing**, the data is made of words which are discrete. This does not allow for backpropagation and thus the Generative Adversarial Nets in this form cannot be applied to this task.
- **Non-convergence to the Nash Equilibrium** - During the training of a GAN, at times, it is possible that the optimum is not obtained. It does not converge to the Nash Equilibrium, but keeps orbiting around optimum.
- **Mode collapse** - Most datasets (such as MNIST, CIFAR-10, etc.) have multiple classes and are therefore multimodal - 10 modes in the case of MNIST. Sometimes, the GAN may not be able to capture all the modes in the training distribution. This results in the phenomenon of mode collapse where only few modes of the data are generated.
- **Vanishing gradient and sensitivity to hyperparameters** - In the algorithm, we train K steps for the discriminator for every step of training of the generator. This value is a hyperparameter which must be tuned to achieve a careful balance between generator and discriminator. Also, if the discriminator trains too fast, the generator gradient vanishes and does not learn.
- **Existence of adversarial examples** - As mentioned in the paper, the issue of adversarial examples - whereby a neural network can classify two images which are perceived by humans as identical, into completely different classes, is a cause of vulnerability in the Generative Adversarial model.
- **Variation in performance on different datasets** - The GAN is observed to perform better on more homogeneous datasets, such as MNIST (which has only grayscale images) than on CIFAR-10 which has colour images.

5 Future improvements

- **Implementation of GANs for text data** - Generative Adversarial networks can be extended to a natural language setting, through some approximations to represent the discrete components of the sentence in a continuous form, so that backpropagation can be done. This is the essence of algorithms such as REINFORCE [1], which have been shown to be successful on text data.
- **Conditional Distributions** - Through ordinary GANs it is possible to generate images irrespective of class. However, it is not possible to get images conditioned on a particular class from the network. A natural solution of this issue is to train a network to condition on particular y values (such as 0-9 in MNIST). This has been implemented in the Conditional GANs paper [2].

References

Some of the ideas in this critique are from the following sources:

[1] GANs for text - https://akshaybudhkar.com/2018/03/26/generative-adversarial-networks-gans-for-text-using-word2vec/

[2] Conditional Generative Adversarial Nets - https://arxiv.org/pdf/1411.1784.pdf

[3] Machine Learning Subreddit - https://www.reddit.com/r/MachineLearning/