Generative Adversarial Networks

Jacob Smith

December 12, 2017

Abstract

Deep generative networks have often fallen short, especial when compared to discriminative models. Since their conception in 2014, generative adversarial networks have seen exponential growth in research. These networks offer promising results for both unsupervised learning and generative models. This research paper presents an overview of generative adversarial networks. Specifically, we will examine the motivation behind GANs, their theoretical justification, the current state of GANs and the research frontiers that have yet to be explored.

1 Introduction

Deep learning have made astounding progress within the past decade. Discriminative models have recently surpassed the abilities of human within the domain of pattern recognition [10]. These successes can attributed to vast, high dimension datasets in conjunction with large neural networks using linear activation functions, dropout regularization techniques and backpropagation to update the parameters [3]. However, deep learning possesses many more ambitious goals. Until recently, success has mostly been seen with supervised classifiers; however, deep generative models now competitively rival their discriminative counterparts.

Generative models learn the joint probability distribution $p(x, y)$ of an input x and label y whereas discriminative models directly learn the conditional probability $p(y|x)$. Generative models may be used as classifiers using Bayes rules to calculate the conditional probability $p(x, y)$ which can then be used to make predictions [8]. Depending on the model, knowledge of the probability distribution can be created both explicitly and implicitly [4]. Those which

Figure 1: The process density estimation of one-dimensional data and a Gaussian distribution [2]. Generative models take a dataset D, sourced from a distribution p_{data}, a create an estimate of that distribution p_{model}.

do not directly model a probability distribution offer mechanisms which require implicit knowledge of the underlying distribution, such as creating a sample from that distribution [4]. As humans, we are able to understand the world around us with tremendous precision. It is easy to underestimate the complexity of the data we process to accomplish this feat. Although progress within the machine learning field is rapidly advancing, computers still have limited understanding of the data the process. Generative models, especially deep generative models, offer promising results towards this goal [6].

The process of training a generative model is very similar to a discriminative model. Using large amounts of data collected from a specific domain, we train the model to generate data from that domain. As these models have fewer parameters than that the number of data samples, they are forced to internalize some representation of the data. Although, unlike supervised training, there exists no explicit desired output. The problem becomes defining a cost function which forces the generative model to produce data more like that of the domain.

Generative adversarial networks (GANs), first introduced in 2014, offer a new framework for estimating generative models with use of an adversarial process [3]. These models offer a clever approach to solving the aforementioned problem of explicitly defining a cost function. Rather than training a single model, a discriminator is introduced. These two networks are pitted against each other in a minimax game. The generative model G attempts to produce data that resembles that of the training set while the discriminative model D attempts to classify whether or not a sample was real of generated. To train this network, G attempts to maximize the error rate of D whereas D attempts to minimize it. Backpropagation is used to update the parameters and to train each model. In the ideal case, D is unable to distinguish the generated samples from the real samples and produces an error rate of $\frac{1}{2}$ [3].

2 Motivations

There exists several compelling reasons for studying generative modeling [2]. These models are extremely useful when your goal is understand the underlying distribution parameters. For example, joint probability distributions of high dimensional data is relevant to both applied math and engineering [2]. Generative models are also applicable to reinforcement learning, particularly model-based learning algorithms. A time series generative model, one which predicts future states of an environment given the current state of the environment and an agents actions, are of particular interest [2]. An agent may learn the best action to take in a given scenario by querying this generative model and choosing the one which outputs the best state of the world. Furthermore, generative models, especially GANs, have recently excelled at semi-supervised learning.

Proposed in the original GAN paper [3], semi-supervised techniques used in conjunction with GANs have recently proven themselves as a reputable technique. The premise involves training the discriminator to classify images into $n+1$ classes where n is the number of class labels and the additional 1 is the class of a fake image. This algorithm is able to obtain great performance on relatively small labeled datasets when compared to labeled datasets typically state of the art supervised algorithms [2]. Currently, the best performing semi-supervised GAN are feature matching GANs, where the new objective of the generative model is to minimize the distance between a layer l of the discriminator [9].

Many applications require multi-modal outputs. For example, predicting the next action of a self-driving car. Within this context, it is important the model not be trained by traditional methods, such as using mean squared error to minimize the distance between the expected and predicted actions. These models cannot be used in situations where there is more than one appropriate prediction. Generative modeling, especially GAN, allow machine learning to function the scenarios where one input corresponds to multiple acceptable outputs [2].

Tasks which involve image generation, modification or translation are all highly applicable to generative models. Although GANs may be used to generate any type of data, images remain the most commonly used source

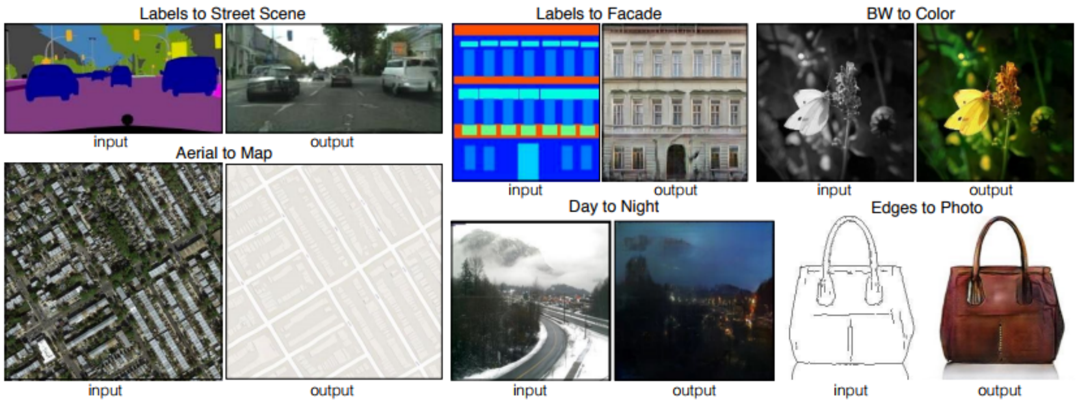

Figure 2: Examples of the image-to-image technique developed by Isola *et al.* [5]. This approach makes use of conditional generative adversarial networks to map image domains.

of training data. Research conducted within the past three years demonstrates the capabilities of GANs to perform these tasks [2, 11, 7]. For example, GANs have recently been shown to be capable of translating aerial photographs into maps and sketches into realistic images [2].

When compared to other generative models, especially deep generative models, GANs possess several advantages... quicker? no Markov chains?

3 Background

Generative adversarial networks may use any differentiable functions for the models; however, the optimal results are typically achieved when neural networks are used [3]. We first define a vector z such that $z\ p_z(z)$. This vector becomes the argument of the generator $G(z; \theta_g)$ where G is a differentiable function and θ_g is its parameters. Another differentiable function $D(x; \theta_d)$ for the discriminator. This function takes as input a vector x, either from the output of G or a sample from the training set, and is parametrized by θ_d. The output of the discriminator is a scalar number representation the probability that the arguments were derived from the training data. In this situation, there exists a minimax game where D attempts to maximize the probability of assigning a correct label and G attempts to minimize $\log(1 - D(G(z)))$. The networks G and D play a minimax game with the value function $V(D, G)$:

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

For the training step, any variant of gradient descent may be used with respect to the individual cost functions [2]. The first updates θ_D while attempting to minimize the discriminator cost function J_D while the other updates θ_G while attempting to minimize the generator cost function J_G. There exist several variations of the cost function that may be used for training purposes; however, the discriminator cost has remained the same for each:

$$J_d(\theta_D, \theta) = -\frac{1}{2}E_{x \ p_{data}(x)}[\log(D(x))] - \frac{1}{2}E_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

In this minimax version of the loss functions, the generator suffers from a vanishing gradient when the discriminator confidently rejects the generated data. The initial solution proposed by Goodfellow was to train the generator to maximize $log(D(G(z)))$ rather than minimize $log(1 - D(G(z)))$ [3]; however, even this new cost function does not completely remove the gradients problems [1].

n: The number of training iterations
k: The number of discriminator iterations
for *training iteration in* n **do**
  for *discriminator iteration in* k **do**
    sample m vectors $\{z^{(1)}, ..., z^{(m)}\}$ from distribution $p_z(z)$;
    sample m vectors $\{x^{(1)}, ..., x^{(m)}\}$ from the training samples;
    update the discriminator using its stochastic gradient

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

  end
  sample m vectors $\{z^{(1)}, ..., z^{(m)}\}$ from distribution $p_z(z)$;
  update the generator using its stochastic gradient;

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} [\log(1 - D(G(z^{(i)})))]$$

end

Algorithm 1: The initial generative adversarial network algorithm developed by Goodfellow *et al.* in 2014 [3]. Multiple variations of this algorithm have since been produced. One of the benefits of adversarial networks is that they may be updated using regular gradient descent.

This algorithm presents the initial training algorithm used by Goodfellow *et al.* when GANs were introduced in 2014 [3]. Many modifications have been made by various sources. The first step in the algorithm is to optimize the discriminator D. Optimally, one would train D to completion; however, this is unrealistic in practice. The number of discriminator training iterations becomes a hyperparameter of the algorithm. For the initial GAN experiments, this number was set to 1 to reduce computation time.

4 Issues

Generative adversarial networks are often characterized by their instability. The training process of

Shape, count, orientation

5 Improvements

Feature matching

6 Model Variations

Deep Convolutional GAN

Wasserstein GAN to improve stability

Reccurent GAN

Conditinal GAN

Growing GANs

7 Research Frontiers

GANs are an exciting idea that are currently leading the frontier of deep generative models.

Hiton's Capsule Network

Improving the stability of GANs

8 Conclusion

Generative models are currently being applied to many problems [6]; however, they offer many more possibilities due to their ability to understand the data they are given.

References

[1] M. Arjovsky and L. Bottou. Towards Principled Methods for Training Generative Adversarial Networks. *ArXiv e-prints*, January 2017.

[2] I. Goodfellow. NIPS 2016 Tutorial: Generative Adversarial Networks. *ArXiv e-prints*, December 2017.

[3] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. *ArXiv e-prints*, June 2014.

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[5] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. *ArXiv e-prints*, November 2016.

[6] Andrej Karpathy, Pieter Abbeel, Greg Brockman, Peter Chen, Vicki Cheung, Rocky Duan, Ian Goodfellow, Durk Kingma, Jonathan Ho, Rein Houthooft, Tim Salimans, John Schulman, Ilya Sutskever, , and Wojciech Zaremba.

[7] M. Mirza and S. Osindero. Conditional Generative Adversarial Nets. *ArXiv e-prints*, November 2014.

[8] Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 841–848. MIT Press, 2002.

[9] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved Techniques for Training GANs. *ArXiv e-prints*, June 2016.

[10] J. Schmidhuber. Deep Learning in Neural Networks: An Overview. *ArXiv e-prints*, April 2014.

[11] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised Cross-Domain Image Generation. *ArXiv e-prints*, November 2016.