ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 11, November 2017

Survey on Generative Adversarial Networks

[1] Mahima Vuppuluri, [2] Abhishek Dash

***Abstract*— Generative Adversarial Networks or GANs were introduced by Ian Goodfellow and his colleagues at the university of Montreal. The concept behind these networks is that, two models fighting against each other would be able to co-train and eventually create a system that could learn more, with less help from humans, effectively reducing the huge amount of human effort required in training and creating deep learning models. GANs are the new class of two different deep neural networks which compete with one another to generate similar data, which leads to the creation of high quality fake information. What is the use of this generated data? If a computer can generate data, then it can use it to understand the scenario it is currently in. GANs are an interesting development in the domain of machine learning and more importantly, unsupervised learning. They can be applied in many fields, from generating text to predicting diseases. They are used for creating images from words, extracting high resolution images from low resolution pictures, currently research is also going on where it is used to generate new molecules which can be helpful in treating an ailment. They have been used also in games for generation of different scenes. Through this paper, we aim to understand what exactly Generative Adversarial Networks are and what are the existing applications of such models. We also consider the existing re-search challenges that exist in this domain and potential use cases that such a system could support.**

***Keywords*- Deep Learning, Semi-supervised Learning, Unsupervised Learning, Neural Networks, Generative Modeling,**

I. INTRODUCTION

Humans learn by looking and seeing the world around them. Experience and observation is integral to human learning. Artificial Intelligence is a field which is witness to unparalled development. Soon artificially intelligent systems would move ahead from basic image and text recognition to more complicated tasks like planning, prediction, reasoning and adapting their systems to the way we think. For a machine to have these functions, we need to provide with with an internal mathematical model of how the world functions., with a huge amount of traning data. Extrapolating on the model and the training data, the machine develops its ability to predict. What is missing today, is the ability of a model to train itself, by generating the training data itself. This is where General Adversarial Networks comes in. In the world of Artificial Intelligence, convoluted neural networks have been successful at training machines for image recognition using supervised data sets, but unsupervised learning remains a black box. There hasn't been much progress it terms of unsupervised learning so far. With deep learning, we can train systems to do more complicated tasks. Instead of having a neural network that can just classify an image, we can develop a system that can generate the image based on some sample content.

Humans minds are very good at predicting things. For example, we know where we need to place our hands to catch a ball coming our way without having to do explicit mathematical computations to come up with the answer. Basically, humans come up with answers by just observing the world around them. Adversarial networks are a new development, which enable machines to predict the outcome consist of a generator, which generates data from a random source of input, and a discriminator, which receives the generated data generated by the generator and also, a real set of data. The job of the discriminator is to distinguish the real from the generated. The two neural networks optimize themselves and eventually the generator produces more realistic images and the discriminator gets better at telling apart the fakes. This leads to the machine having a better idea of what is possible in the real world.

II. WHY GANS ARE WORTH RESEARCHING

Adversarial training is complicated because both the generator and the discriminator need to be optimized simultaneously. Such optimization is difficult and if the model isn't stable, the system will not work as desires. It was thought till now that General Adversarial Networks were very unstable. Many times, the generator does not learn to produce what we understand as good images. A lot of research has gone into stabilizing these networks. The papers propose Laplacian Adversarial Networks(LAPGAN) and Deep Convolutional Generative Adversarial Networks(DCGAN), and Adversarial Gradient Difference Loss Predictors(AGDL), which can also perform video generation. These systems predict plausible situations of the world regardless of the kind of videos and images provided to them.

This adversarial training is different from traditional neural networks in one discerning way. Every neural network has a cost function — a function that tells us how well the network is functioning. This cost function forms the basis how well a neural network learns and what it learns. A traditionally neural networks are given a cost

ISSN (Online) 2394-2320

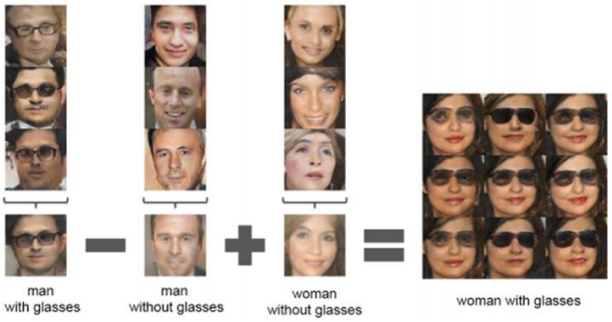
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 11, November 2017

function that is carefully constructed by a human scientist. For complex processes, such as generative models, developing an optimized cost function is not an easy task. This is where the adversarial network shine. The adversarial network manages to learn its own cost function — its own complicated rules of what is right and what is wrong — overcoming the need to carefully design and build a cost function.

In practice, this is the property of adversarial networks that translates to sharper, better generation of data and higher-quality predictions. To show this, LAPGANs and DCGANs have been trained on a large quantity of image data sets capturing either a specific set of images such as bedrooms, faces, and paintings, or a set of images from the ImageNet data set that are diverse regular images. DCGANs have become very popular in the Machine Learning and artificial Intelligence community, and several coders have taken the code that has been released for DCGANs and trained and created newer variants of the DCGAN on various sets of data. For example, here's a DCGAN trained by researchers from NVIDIA on 18th-century paintings.

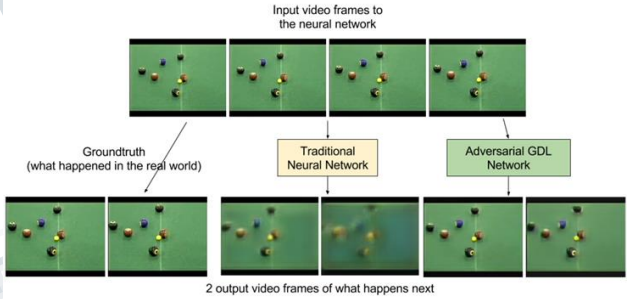
DCGANs are also able to figure out patterns and put similar representations together in some dimension space. For example, in the faces data set, the generator doesn't understand what smiling means, but it is able to group together images of smiling people as it is able discern similarities in them.

This demonstration of the ability of unsupervised generative models to lear object attributes like rotation, position, scale, and semantics was one of the first of its kind.



Once a machine is trained to predict what the world is going to look like, we can apply its knowledge to different tasks and scenarios. If we take two images, one with the camera moved slightly to the right for a different viewpoint, the machine will notice the differences and learn that the environment is three-dimensional. Then, it could apply that observation to a new situation, such as figuring out the distance someone would need to reach to grab a pen from across the table. As machines acquire this common sense, they become better at knowing the best approach to do specific tasks, and can quickly reject any hypotheses that would be implausible in the real world, such as an attempt to walk through a door without opening it first. Ultimately, this type of knowledge could help further accelerate the development of applications including advanced virtual assistants and chat bots.

With AGDL, (Adversarial Gradient Difference Loss Predictors) we progress further toward this goal of predicting what will happen next. We take a few frames of video and build models that will be able to predict what will comes next. For a game of snooker, for example, that means forecasting the future frames of the cue balls moving, as soon as the shot is hit.



While AGDLs have been a first significant step in forward video prediction, we see the need for considerable set of improvements to our current models, so that they can predict further into the future, before we can start reasoning over the predictions of these models and using them to plan.

GANs are used for inverse reinforcement learning and imitation learning. This could be applied to understand the way individuals drive vehicles by either replicating human drivers exactly or inferring the conclusions that human drivers make and learning policies that accomplish the same decisions in a better way. Studying the imitation of human drivers using GANs can be better than learning to use human drivers with traditional supervised learning because GANs can handle situations where multiple correct actions are present. GANs can be used for generating realistic situations in simulation for training self-driving car policies. Apple did something along the same lines, though not for self-driving cars: Apple's first

ISSN (Online) 2394-2320

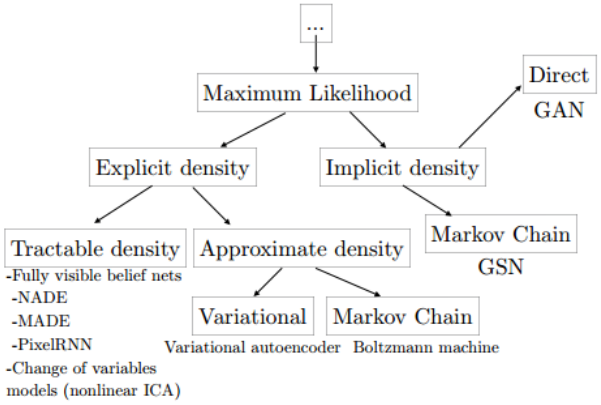
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 11, November 2017

ML research paper focuses on machine vision. GANs can learn to re-render a scene from a different point of view, which could be useful for making self-driving cars that can work with a wide variety of camera configurations and locations. GANs could be used to standardize machine learning systems and make them more accurate, kind of like what Tim Salimans did for semi-supervised learning in [1] Improved Techniques for Training GANs.

III. OVERVIEW OF GANS

Supervised Machine Learning is basically deducing a function from labelled training data where each input has a specific output.
Unsupervised Machine Learning belongs to the class of problems where for each input data no labels are there as an output. Hence the underlying hidden structure in the data is figured out by the algorithms.
Semi Supervised Learning belongs the class of problems for which we have labels for some training data and some are missing. Neural Networks are one of the computing structures that can be used to solve these class of machine learning problems.
Neural Networks are a type of computing structures that mimic biological neural networks. They consist of neurons (similar to real neurons) connected to each other in a network like structure (comparable to synaptic connections in human brain) with different layers of neurons. Neural Nets have huge advantage when dealing with intricate data. They can learn adaptively and make deep insights on data which might be incomprehensible for humans.

B.Historical Overview



Taxonomy of Generative Models (from Ian Goodfellow's NIPS tutorial, 2016)[3]

Now that we have laid the ground on what generative modeling is, and why they are useful, let's look at the various approaches in generative modeling. For comparing the models, they can all be described to perform maximum likelihood.

The above graph depicts the various families of generative models as described by Ian Goodfellow in his NIPS tutorial. Let's now take a look at the pros and cons of some of the popular approaches in the families of models stated above.

Fully Visible Belief Networks

Fully Visible Belief Networks are known to use the chain rule of probability to decompose the probability distribution over a vector into a product by multiplying each members of the vector. The formula is as follows

$$p_{\text{model}}(\boldsymbol{x}) = \prod_{i=1}^{n} p_{\text{model}}\left(x_i \mid x_1, \ldots, x_{i-1}\right).$$

The most well knows model in this family is PixelCNN.

Images generated by PixelCNN are shown in the picture above. The biggest drawback with Fully Visible Belief Networks is that the rate of at which samples are generated is very low. Every time a new sample is requires, the model needs to be run again. This cannot be done in parallel.
Such models are based on change of variables. They begin with a simple distribution like a normal distribution and use a non-linear function to transform this distribution to a different space. The main issue with this is that the transformation needs to be designed to be reversible, and the latent variables must have the same dimensionality as the data. So, if you want to generate 1000 pixels, you need to have 1000 latent variables.

Variational Autoencoder

The way Variational autoencoders have encoder, decoder and a loss function. The encoder reduces the dimensions of the data and tries to compress it without huge loss in data whereas the decoder tries to check how much

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 11, November 2017

information is lost when the data is reconstructed. The loss function for a data point xi

$$l_i(\theta, \phi) = -E_{z \sim q_\theta(z|x_i)}[\log p_\phi(x_i|z)] + KL(q_\theta(z|x_i)||p(z))$$

The loss function helps decoder to learn to reconstruct the data. It tries to maximize the lower bound on the log likelihood of the data.

The main disadvantage with the model is that it is asymptotically consistent if the distribution q is perfect. Otherwise, there would be a gap between the lower bound and actual density of the data. Another disadvantage is that the samples generated are of relatively low quality

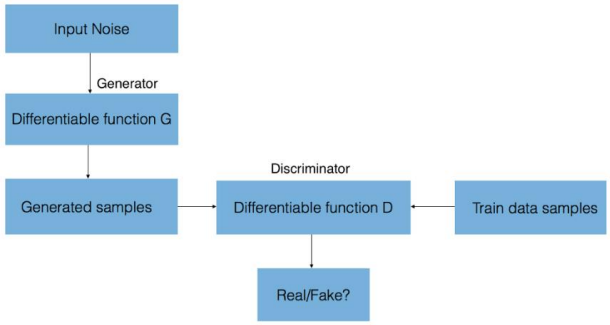
Images of celebrity-like faces generated by a VAE (By Alec Radford)[5]

.Boltzmann Machines

A Boltzmann machine can be well-defined using an energy function, and a state's probability is proportional to each of the energy's value. Renormalization is performed by dividing sum over different states for the conversion to an actual probability distribution. Boltzmann Machines are used to find complex regularities based on simple learning algorithm[7]. "A surprising feature of this network is that it uses only locally available information. The change of weight depends only on the behavior of the two units it connects, even though the change optimizes a global measure" [7]. This is an intractable sum, which demands for approximation using Monte Carlo methods. The drawbacks are these Markov Chain Monte Carlo methods don't do well for high dimensional spaces. So they won't work well on images from ImageNet as well as they perform on images like MNIST.

C.Technical Overview

Generative Adversarial Networks belong the class of neural networks having a combination of two models (neural networks) namely Generative Models and Adversarial Models.



GAN overview

These models are used to estimate maximum likelihood (semi supervised learning).

An adversarial model is used to learn conditional Probability function(P(y|x) i.e. given set of values 'x' (observed values), we can predict 'y' (unobserved values). Unlike adversarial models, a generative model learns joint Probability Distribution function f(observation data, target/label values) from some input data that can be used to generate fake data. Also, using Bayes Rule it can be converted to conditional Probability.

There are two types of generative models: 1) Implicit Density Models 2) Explicit Density Models.

With Explicit Density Models one needs to express a model for generated samples(pmodel(x)) explicitly and then solve for it whereas in Implicit Density Models one needs to learn model that can take samples from (pmodel(x)) without explicitly defining it. GANs belong the class of Implicit Density Models.

The main logic behind GANS is two models namely generator and discriminator competing against each other in a zero-sum game. The generative model generates samples based on the underlying distribution as the training data. The discriminator tries to distinguish/differentiate between the samples generated by the generator and real data from training set.

A common analogy of this game is that of a forger and the police. The forger is like the generator, trying to counterfeit money, and making it look as legitimate as possible to fool the police. The police are like the discriminator whose goal is to be able to identify money that has been counterfeited.

[one image explaining structure of GANs]

the difference between discriminative and generative algorithms are: discriminative algorithms learn the boundaries between classes (as the Discriminator does) while generative algorithms learn the distribution of classes (as the Generator does).

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 11, November 2017

In order to learn the Generator's distribution, p_g over data x, the distribution on input noise variables p_z(z) must be defined. Then G(z, θ_g) maps z from latent space Z to data space and D(x, θ_d) yields a single scalar—probability that x came from the real data rather than p_g.
The Discriminator is trained to maximize the probability of assigning the correct label to both examples of real data and generated samples. While the Generator is trained to minimize log(1—D(G(z))). In other words—to minimize the probability of the Discriminator's correct answer.
It is possible to consider such a training task as minimax game with value function V(G, D):

$$\min_G \max_D V(D,G)$$

$$V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z))]$$

IV. TYPES OF GANS

A. DCGANs

DCGANs were the initial main development on the GAN architecture. They are more steady in terms of generating higher quality samples and training.
The writers of the DCGAN engrossed themselves on improving the architecture of the original vanilla GAN spending quite a long time doing the most stimulating thing about deep learning. The major things that they found out was:
•Batch normalization must be done in both the networks.
•It is not a good idea to have fully hidden connected layers.
•One must stride with convolutions and avoid pooling.
•Your friends are ReLU activations.
DCGANs have become one of the reference points to start understanding and implementing GANs. DCGANs should be used if one wants to use something better than vanilla GANs .One should always compare their own new GAN algorithm with DCGANs(as a solid baseline) since Vanilla GANs work on simple datasets.

Improved DCGANs:

There were still several problems with DCGANs even after a series of improvements. Hence a series of methods to improve upon DCGANs have been suggested in [1]. This results in better generation of high-resolution images and stabilizes the convergence somewhat which was one of the main issues Following are some of the enhancements proposed by the authors:
• Feature matching: An objective function is proposed as an alternative to the competition between generator and discriminator to fool each other as much as possible. The objective function forces generator to create data that is similar to the real data. And the discriminator only has to specify which statistics are worth matching.
• Historical averaging: Historical values are considered when we are updating the parameters.
• One-sided label smoothing: Here they make the discriminator's target output from [0=fake image, 1=real image] to [0=fake image, 0.9=real image] which really improves your training.
• Virtual batch normalization: Using statistics that we collect on reference batch we can avoid depending on the same batch data. We use it on generator only as It is computationally expensive.
All of these improvements allow the model to perform very well at producing high resolution images which is one of the weakness of GANs. Following 128x128 images is a comparison between DCGAN and improved DCGANs to understand the subtle differences

DCGAN | Improved DCGAN

All of these images are supposed to be of dogs. DCGANs fail miserably at representing them but with improved DCGANs at least one can see a distorted representation of a dog. This also demonstrates another lacuna of GANs i.e. creating structured content.

B. CGANs

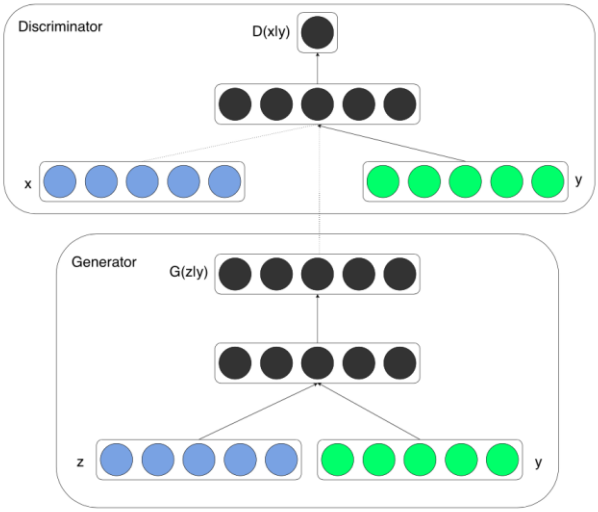
We have two neural networks in GANs: the generator G(z) and the discriminator D(X). If we add a vector y to both the networks to condition them, then it is an easiest way to condition when we feed y in both networks. So our generator and discriminator become G(z,y) and D(X,y) respectively
We can look at it from a probabilistic view:We can model our data distribution G(z,y) given z and y i.e. our data is produced with the following scheme: X~G(X|z,y).
Similarly for the discriminator, now it attempts to discover discriminating label for X and XG, which are modelled with d~D(d|X,y).
Hence, we could see that both D and G is jointly conditioned to two variables z or X and y.
The objective function is:

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 11, November 2017

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

[8]. When we compare above objective function with GANs we can see that an additional parameter y only has been added. But in the GAN's architecture we have an additional layer for both discriminator net and generator net. The architecture of CGAN is given in the following diagram.



C. InfoGANs

InfoGANs, an information-theoretic extension to the Generative Adversarial Network that can learn disentangled representations in a wholly unsupervised manner[6]. InfoGANs produce interpretable representations for latent variables in the model by building upon the GANs framework. The crucial idea is to fragment latent variables into a set c of interpretable ones and a source of uninterpretable noise z. Interpretability is stimulated by addition of an extra term in the original GAN objective function which captures the shared information among the interpretable variables c and the output from the generator. The InfoGAN minimax optimization is defined as:

min G max D VIG(D, G) = V (D, G) − λ • I(c; G(z, c))

where G refers to the generator, D refers to the discriminator, z refers to the uninterpretable noise, c encodes the salient latent codes and the Mutual Information (I) is given by: I(c; G(z, c)) = Entropy(c) − Entropy(c|G(z, c)) [6]. The uniqueness of the InfoGAN objective function compared to the regular GAN one is the introduction of a regularization term which involves mutual information between the latent codes c and the generator G. The second entropy term needs admittance to the posterior p(c|G(z, c)) which is approximated by the discriminator network. mutual information can be exploited whenever we are concerned in learning a parametrized mapping from input X to a higher level representation Y that conserves information about original input. If we are concerned in a mapping q(Y |X; θ) that conserves information about X, where θ are the parameters to be learned, this can be attained by maximizing the mutual information between X and Y , I(X; Y). This is denoted by infomax principle and in the case of InfoGAN interprets to maximizing the common information between the latent codes c and the output from the generator model, G(z, c). The job of maximizing common information is comparable to training autoencoder and minimize reconstruction error. This implies that learning a mapping that ties c with G(z, c) can be understood as combining or encoding c as thoroughly as possible in the output G(z, c).The codes in c aid as a label for the outputs produced from G.

V. CONCLUSION

In this paper we have attempted to compare and understand the applications of different GANs and how they work. We found that DCGANs can be used to create high resolution images. With the help of Conditional GANS one can control the generation of images with its conditional variable. InfoGANs are used if datasets are not that complex and one doesn't have label data. They can decipher the meaningful image features of dataset.

Adversarial networks offer a robust algorithmic context for building unsupervised/semi-supervised learning models that combine properties such as common sense which are a key part of Human Brain, and we believe that enduring to explore and push in same direction gives us a realistic chance of succeeding in our pursuit to build smarter AI.

REFERENCES

[1] Improved Techniques for Training GANs, Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen

[2] Chen, Xi, et al. "Infogan: Interpretable representation learning by information maximizing generative adversarial nets." Advances in Neural Information Processing Systems. 2016.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in NIPS, 2014, pp. 2672–2680

[4] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in NIPS, 2014, pp. 3581–3589.

[5] Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, Alec Radford, Luke Metz, Soumith Chintala

[6] InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, Pieter Abbeel.

[7] A Learning Algorithm for Boltzmann Machines, David H. Ackley Geoffrey E. Hinton, Terrence J. Sejnowski

[8] Conditional Generative Adversarial Nets,Mehdi Mirza, Simon Osindero