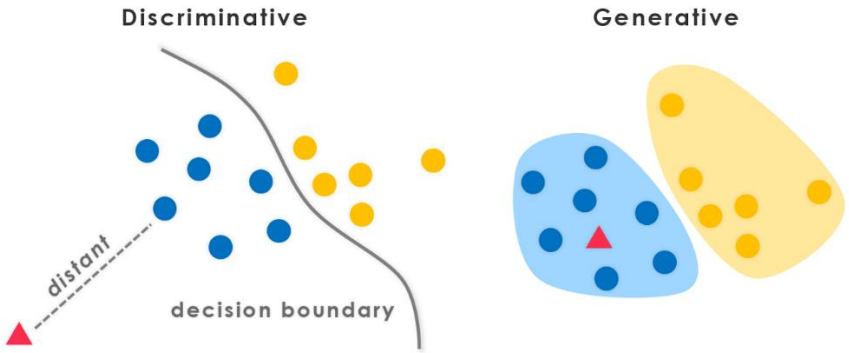
Understanding Generative Adversarial Networks

By: Arjun Mishra

Introduction to Generative Models

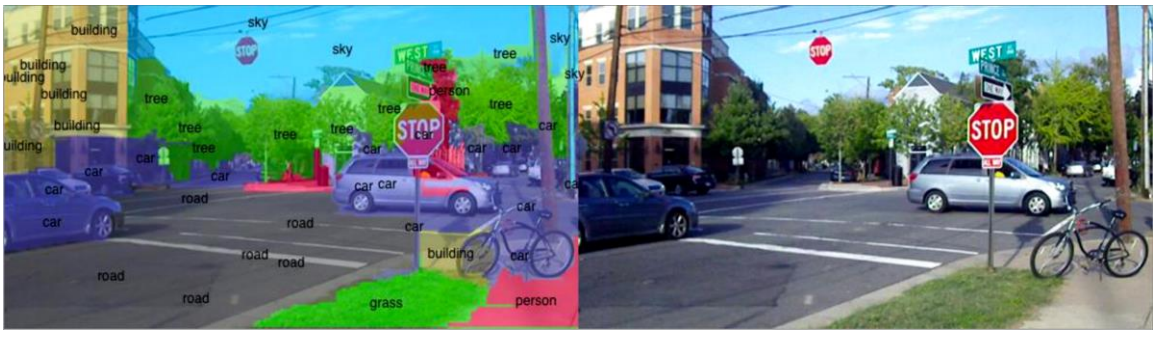
Machine learning models can be broadly classified into generative and discriminative models. While a discriminative model tries to model the conditional probability of a target variable based on the given data, generative models try to arrive at the joint distribution of the whole data. Both classes of models have their own advantages and disadvantages. One of the major advantages of generative models over discriminative models is being able to generate data from the joint probability density function. Due to this, generative models have found application in various areas where "creation" of an output is key.

Figure 1: A general idea of the distinction between a discriminative and generative model



The unsupervised nature of learning for these kinds of models is what makes them highly important in the quest for solving general intelligence. Generative models have been identified as crucial in the development of AI systems that can automatically build understanding from raw input. For example, if we feed a picture of a car to a computer, it does not know what it represents or means. To the computer, it is just a matrix with some numbers. Drawing up a parallel of this situation from the digital world in the real world, we can say that when a child looks at a car for the first time, it does not know what it is either. After observing it in action again and again (and being told what it is), the child figures out what it is, how it works and can also successfully classify other objects that look like it, as a "car". In the same way, we want the computer to understand objects from the real world automatically. Even if we explicitly don't provide labels for those objects to the machine, it should be able to reproduce objects of a certain kind automatically based on its understanding.

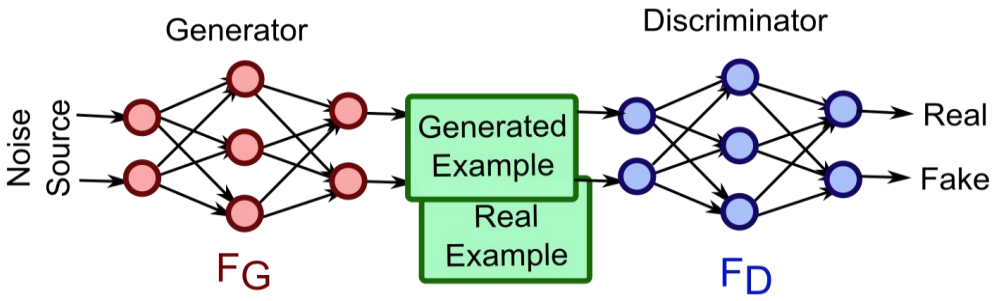
Figure 2: An example of deep learning models labeling real world images. Image is taken from the ImageNet set



Generative Adversarial Networks: Background and Functioning

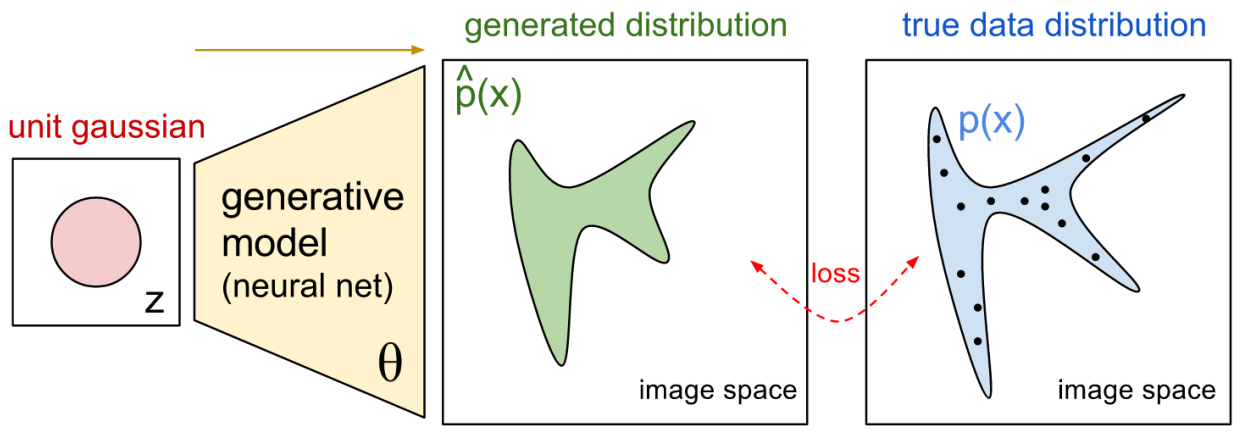
Generative Adversarial Networks (GANs) are a major step towards this kind of understanding. A brainchild of Ian Goodfellow, GANs have changed the way how everyone approaches building generative models with machine learning (there is some debate over whether the idea was first introduced by Jurgen Schmidhuber. Ian Goodfellow and Jurgen will write a paper together that highlights the differences in their ideas). First introduced in 2014 in a paper by Goodfellow and his colleagues at the University of Montreal, they have been hailed by some pioneers in the field of deep learning as one of the biggest development in the field of machine learning in the last 10 years. By using a combination of two neural networks competing against each other in a zero-sum game, GANs have been able to generate original photo-realistic pictures. Why the competing networks? In the attempt to outsmart each other, they collectively perform a lot better. The first network is obviously a generator network which takes as input a set of random numbers and outputs an image using components of a deconvolutional neural network like deconvolutional layers. If we feed some random numbers into such a network, it will produce an absolutely random image. However, we want the model distribution to match the true data distribution in the space of images. Suppose as an example we only want it to create images of cars. This is where the GANs approach introduces a clever trick in the form of a second network, discriminative in nature.

Figure 3: The two networks working in a GAN



The second network takes as input the randomly generated images from the first network, labeled as fake and externally provided real images, labeled as real. Its objective is to correctly identify and classify the generated car images from the real ones. When working in tandem, the loss function of the generative network will be configured in a way that there is maximum loss when the discriminative network can classify all the images correctly. Thus, backpropagation through the networks will try to tune the parameters of the generative network to generate images that make it hard for the discriminator to classify as generated. However, at the same time, the discriminator is trying to minimize its loss too. This back and forth interaction between the two networks results in the creation of images which are indistinguishable from the real ones and in the creation of a network which is an expert in identifying real images from fake ones.

Figure 4: A high level illustration of a Generative Adversarial Network

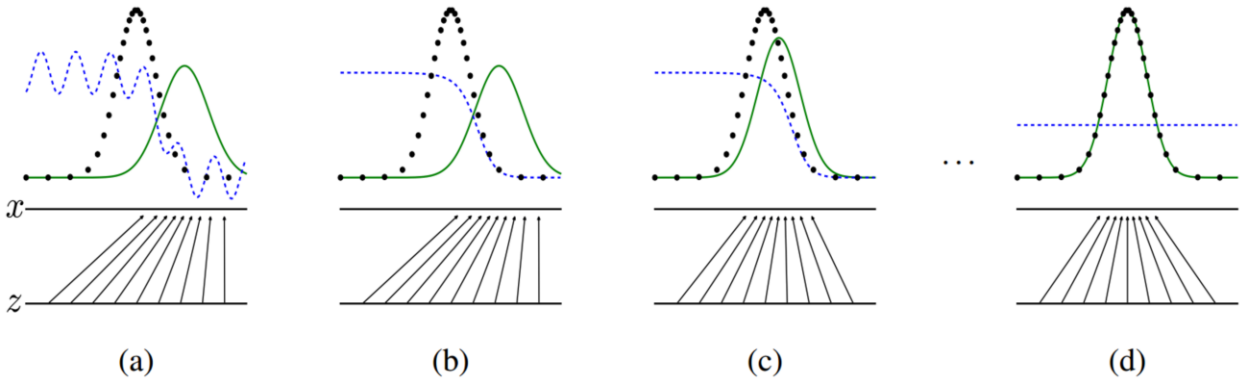


Implementing such a model on the ImageNet dataset which contains millions of images or in other words, millions of "representations of our real world" can enable a machine to learn the structural nuances of our world. The generative network will be able to recognize the various symmetries, geometries and complexities of objects usually found in our world. Speaking in strictly technical terms, how GANs do this boils down to the tuning of the weights. Thus, the network must capture a vast amount of information in the form of the most optimized weight matrices. Translating this to the real world, a compression of this magnitude will result in capturing the starkest features and relations in the construction of our world.

Now that we have discussed how GANs work theoretically, we can start to discuss some of the math behind the idea. We will denote the true distribution of the data as p_{data}. Let $G(z;\theta_g)$ denote the generative model. In this case, G is restricted to be a neural network with parameters θ_g that takes as input some noise z with distribution p_z and outputs a sample per G's distribution, p_g. We define the second (discriminative) neural network $D(x;\theta_d)$ that outputs a single scalar representing the probability that x came from the data rather than p_g. The GAN approach simultaneously trains both G and D. The discriminative model D is trained to maximize the probability of assigning the correct label. Thus the probability output of D should be high when the sample comes from p_{data} and low when the sample comes from p_g. Additionally, the generative model G is trained to fool the discriminative model into believing samples from p_g are actually from p_{data}, thus G's goal is to maximize the probability output of D when the sample comes from p_g. This leads to the following minimax function:

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

Figure 4: Steps in training a Generative Adversarial Network



The training procedure alternates between updating D whose distribution is visualized by the blue dashed line, and G which maps the space of z (lower horizontal line) to the space of x (upper horizontal line); the distributions p_{data} and p_g are visualized by the dotted black line and solid green line, respectively.

Starting at (a), the generative model G is near convergence and D is a partially accurate classifier.

In (b), the discriminative model D is updated, tending towards the optimizer: $D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$

(c) Once D has been updated, the gradient for G changes to flow to regions where D classifies as data.

(d) After convergence, the optimal result is to have p_g perfectly mimic the data distribution and D to be unable to differentiate between the two distribution, i.e. $D(x) = 1/2$.

We now know about the working of GANs. Next, we will discuss some of the areas where GANs are being applied today.

Applications of GANs

From our description of GANs so far, we can immediately think of applications centered around generating various kinds of data, including images and sound, from unknown and complex distributions. Around this main concept, there are a lot of applications that stem from it after implementing some tweaks on top of it. We will list out and discuss some of these cases:

- Generating a high-resolution image from a low-resolution image – GANs have proved to be much better than Recurrent Neural Networks in creating a high-resolution image from a low-resolution

one. Using GANs we can thus recreate high-res images that are not available to us in high quality (eg. Those taken from cameras in the past) with minimal manual effort.

- Ian Goodfellow lists out a few current and potential areas where GANs could be useful in the upcoming field of self-driving cars:
    - GANs can learn to re-render a scene from a different viewpoint, which could be useful for making self-driving cars that can work with a wide variety of camera configurations, etc.
    - GANs can be used for generating realistic environments in simulation for training self-driving car policies. At present, OpenAI's platform Universe is being used to train self-driving cars where the environment from the game Grand Theft Auto is being used to simulate the real world.
    - GANs are used for imitation learning and inverse reinforcement learning. This could be applied to studying the way humans drive and either imitating human drivers exactly or inferring the goals that human drivers have and learning policies that accomplish the same goals better. Learning to imitate human drivers using GANs can be better than learning to use human drivers with traditional supervised learning because GANs can handle situations where there are multiple correct actions.
- Generating original art, music or design based on an underlying theme: The generator in GAN will try to replicate whatever images we feed into the discriminator but also change it in a unique way. This property has found use in creating original art inspired by the designs of famous artists like Van Gogh, Michelangelo etc. However, we can see the extension of this attribute to other areas like creating music from a particular genre and imitating the style of an artist. Also, we can use GANs to come up with new designs in areas like automobile design, architecture etc. These networks are not limited by human imagination and can produce truly novel creations.
- Generating images from text descriptions: A spinoff of the previous application is generating images from text descriptions. For example, if our GAN knows what a blue jay looks like and what the color white looks like based on our supervised training samples we feed it, we can ask it to generate the image of a white ("blue") jay. Imagine having the power to generate multiple options of an idea – such a product could revolutionize industries like the entertainment, art etc.
- Image completion: Don't have access to a photoshop expert? GANs to the rescue again. If there are parts of an image that are clipped or missing, GANs can be used to make those images whole.

Shortcoming of GANs: The most glaring shortcoming of GANs is the reproduction of training samples that we have fed to the discriminator network. This is the problem of non-convergence. It can happen in a lot of cases that the Generator-Discriminator battle does not reach equilibrium. If we don't use a large enough dataset and train the network for a long time to see some favorable results, the network will surely end up overfitting. We will start to see the same output as the inputs we fed to the discriminator

network. If we think about it, this was the task that we set out to accomplish – having an expert generator that creates, for example, images that are extremely hard for the discriminator to classify. However, we wanted to create samples that were original and could fool the discriminator, not exact replicas. This means that the generator has not understood the salient features of the data we fed it but has instead tuned its parameters to arrive at the best result it can possibly achieve. If the dataset we feed it is big enough, however, the network will be forced to develop a deeper understanding of the data and hence create original outputs.

GANs are still a relatively new model in the world of machine learning and there is continuous ongoing research in the area. The future for this model is definitely bright!

References

- https://medium.com/@ageitgey/abusing-generative-adversarial-networks-to-make-8-bit-pixel-art-e45d9b96cee7
- https://blog.openai.com/generative-models/
- https://www.quora.com/What-could-be-the-applications-of-generative-adversarial-networks-GANs-in-autonomous-vehicles-if-any
- https://arxiv.org/abs/1701.00160
- http://wiki.ubc.ca/Course:CPSC522/Generative_Adversarial_Networks