

Generative Adversarial Network (GAN)

CMPE-255

Term Project Report

San Jose State University

By

Anusha Velumani, Sahana, Purva Deekshit, Sachin Guruswamy, Nikhil Saunshi

Introduction:

GAN is an unsupervised learning technique of creating images from scratch. It is different and difficult as compared to other deep learning models, because rather than classifying an image, it creates the image based on input parameters.

Currently a lot of research is going on applications of GANs, which include:

Gaming applications to generate a new level of a video game
Generating high resolution image from a low resolution one
Generating text paragraphs like we find in articles
In medical science for generating a new drug with the help of existing drugs and also to predict if a given combination of medicines will be able to cure a disease.

We have tried to explain this concept of GANs and how it works, using an example of high-resolution image generation from input text description.

Here, input is the only text description of the object which we want to generate image for. For example, input will be "This small bird has bright yellow tail and grey feathers". Using this, the model will generate an image matching this description.

How GANs work:

The two main components of GANs are the generator and discriminator which are two deep networks.

Generator:

First, we sample some random noise z using a normal or uniform distribution. Normal distribution is from $(0, 1)$ and Uniform distribution is from $(-1, 1)$. With z and text description as an input, generator G creates an image of x ($x=G(z)$). This will be some random distorted image, which won't make any sense.

Conceptually, z represents the hidden and vague features of the images generated, for example, the color and the size of the bird. At this point, we don't know which bytes represents which features. But as the training progresses, we can see that the generator creates more realistic image, close to real images. This is provided as input to the discriminator.

Discriminator:

The generator alone will only create random images based on the text input. That's why discriminator D is needed. By training on real images and correct input text, GAN builds a discriminator to learn what features make the image real. It looks at the real images (training samples which it has learnt) and images created by generator and tries to

distinguish whether the input image to the discriminator is real or fake. The output of discriminator, $D(X)$ is the probability that the input x is real, i.e. $P(\text{class of input} = \text{real image})$.

Discriminator is trained like a deep network classifier. If the input is real, output will be $D(x) = 1$. If it is generated, it should be 0. Through this process, the discriminator identifies features that contribute to real image.

On the other hand, generator tries to create images with $D(x) = 1$, which are close to real image. So, the decision from discriminator is back propagated as a feedback to generator, so that it will get a direction in which to generate image in the next iteration. The training layers of GANs are deep convolutional layers, each performing multiple transposed convolutions to up sample z to generate the correct image x . In this way, the discriminator guides the generator to produce more and more realistic images.

Eventually, the discriminator can plot very less differences between the real and generated images and the generator creates images which discriminator cannot distinguish from real. At this stage, the GAN model eventually converges and produces a natural looking image.

To measure the error in generated image, a cross-entropy function is used, given by: $p \log(q)$. For real image, p (the true label for real image) equals 1. For generated images, it is $(1 - p) \log(q)$.

G and D simultaneously tries to minimize and maximize the error function: $\min_{\max} V(D, G) = xy$, where G tries to minimize the difference between generated and real image and D tries to maximize it.

GANs also support a feature called Style Mixing. We can select the features which we want to be definitely present in the image and which we want to eliminate from the image.

The coarse features covering high level aspects such as age, hair style and glasses are prominently retained. Middle scale features, such as eye color and other facial features are changed with a little variation and fine features just change small details such as hair style, skin complexion etc. Currently GANs are learning to retain tiny features as well.

We have taken an example of how a two stage GAN synthesizes high resolution image from the text description given as input. Here, first stage GAN uses G and D to up sample the noise Z and produce an image that matches the text description. However, this image is low in resolution.

The second stage GAN down samples this image and using the text description as input, it produces a high-resolution image as the output.

We have shown examples in the presentation of how GANs work. Training dataset includes images of flowers with their correct text description. Input for generating a new image is sentence of words. The model is conditioned on text features encoded by a hybrid character-level convolutional recurrent neural network. And final image is produced.

This is how GANs are trained and used to generate images on their own. The discriminator concept can be applied to existing deep learning applications also. We can add a layer of discriminator into existing deep learning network to provide feedback and improve accuracy.

Why GANs are important:

- Generation of images (Sampling) is straightforward.
- Robust to Overfitting since Generator never sees the training data.

References:

[1] Generative Adversarial Text to Image Synthesis Scott Reed, Zeynep Akata, Xincheng Yan, Lajanugen Logeswaran REEDSCOT1 , AKATA2 , XCYAN1 , LLAJAN1 Bernt Schiele, Honglak Lee SCHIELE2 ,HONGLAK1 1 University of Michigan, Ann Arbor, MI, USA (UMICH.EDU) 2 Max Planck Institute for Informatics, Saarbrücken, Germany (" MPI-INF.MPG.DE)

<https://arxiv.org/pdf/1605.05396.pdf>

[2] GAN—What is Generative Adversary Networks GAN?

https://medium.com/@jonathan_hui/gan-whats-generative-adversarial-networks-and-itsapplication-f39ed278ef09

[3] StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks Han Zhang, Tao Xu, Hongsheng Li, Shaoqing Zhang, Xiaogang Wang, Xiao lei Huang, Dimitris Metaxas (Submitted on 10 Dec 2016 (v1), last revised 5 Aug 2017 (this version, v2))

<https://arxiv.org/abs/1612.03242>

[4] GAN 2.0: NVIDIA's Hyper-realistic Face Generator

<https://medium.com/syncedreview/gan-2-0-nvidias-hyperrealistic-face-generator-e3439d33ebaf>