Generative adversarial network

Refs

1. Generative adversarial networks

   [https://arxiv.org/pdf/1406.2661.pdf

1. A review of Generative Adversarial Networks (GANs) and its applications in a wide variety of disciplines - From Medical to Remote Sensing

   https://arxiv.org/pdf/2110.01442.pdf

Generative adversarial networks

GANs are the most interesting idea in the last 10 years in machine learning

- Yann LeCun

- **Generative** models can generate new data instances.
- **Discriminative** models discriminate between different kinds of data instances.

Generative algorithms

GANs belong to **generative algorithms**:

- Generative algorithms and discriminative algorithms are two categories of machine learning algorithms.
- If a machine learning algorithm is based on a fully probabilistic model of the observed data, this algorithm is generative.
- Generative algorithms have become more popular and important due to their wide practical applications.

Generative algorithms

Generative algorithms can be classified into two classes:

- Explicit density model
- Implicit density model

Explicit density model

An explicit density model assumes the distribution and utilizes true data to train the model containing the distribution or fit the distribution parameters.

When finished, new examples are produced utilizing the learned model or distribution.

Explicit density model

These explicit density models have an explicit distribution, but have limitations.

For instance, learning is conducted on true data and the parameters are updated directly based on the true data, which leads to an overly smooth generative model.

The generative model learned by approximate inference can only approach the lower bound of the objective function rather than directly approach the objective function, because of the difficulty in solving the objective function.

Implicit density model

An implicit density model does not directly estimate or fit the data distribution.

It produces data instances from the distribution without an explicit hypothesis and utilizes the produced examples to modify the model.

Prior to GANs, the implicit density model generally needs to be trained utilizing either ancestral sampling or Markov chain-based sampling, which is inefficient and limits their practical applications.

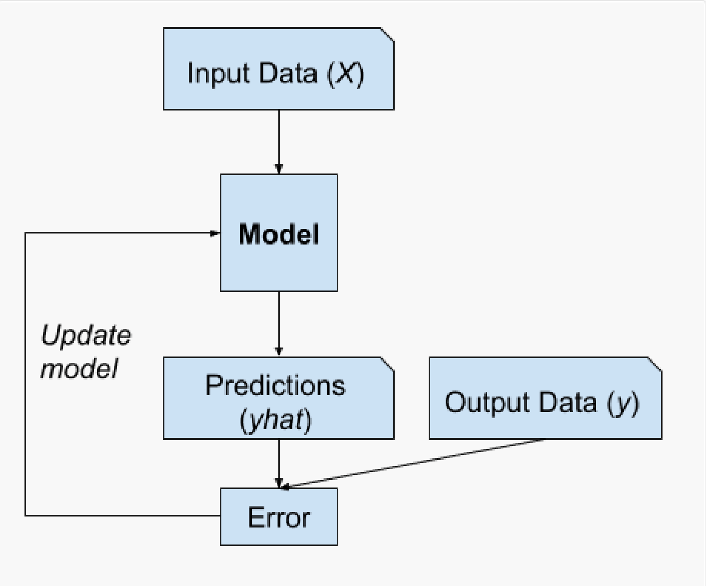
GANs belong to the directed implicit density model category.

GAN

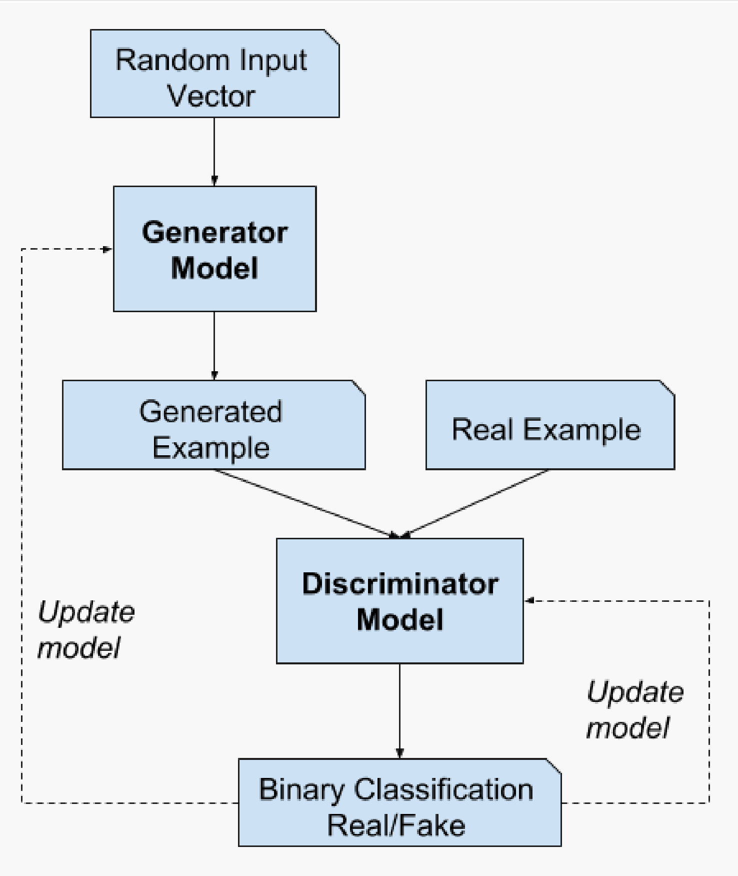
proposed by Ian Goodfellow of Google Brain scientists in 2014.

GANs are structurally inspired by two-person zero-sum games in the game theory (i.e. the sum of the two people interests is zero, and the gain of one side is exactly what the other side loses).

It sets up one generator and one discriminator for each participant in the game.



Example of Supervised Learning



Example of the Generative Adversarial Network Model Architecture

- The aim of the generator is to learn and capture the potential distribution in the actual data samples as much as possible, and generate new data samples.
- Discriminator is a binary classifier, and the aim is to determine whether the input data is from the actual data or from the generator.
- In order to win the game, the two players need to constantly improve their capability to generate and discriminate. So the process of optimal learning is a minimax game problem.
    - The aim is to find a Nash equilibrium between the two sides, so that the generator can estimate the distribution of data samples.

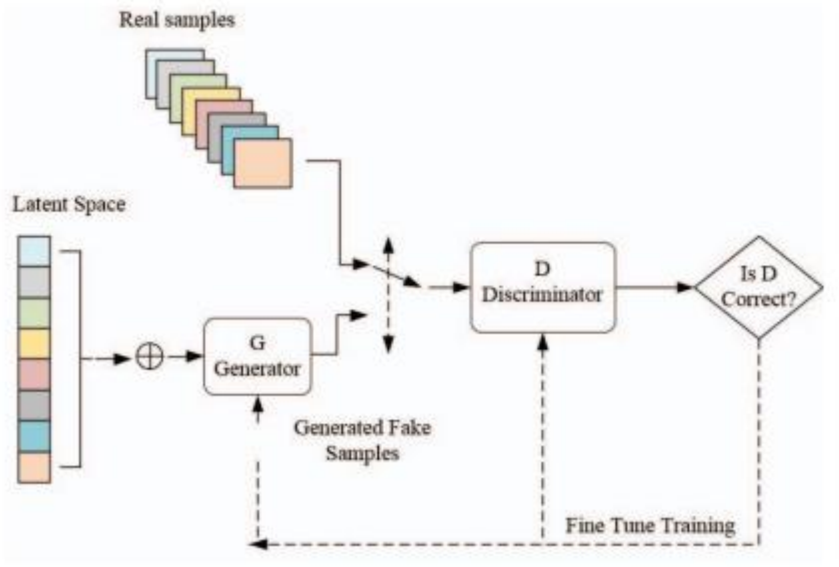
Nash equilibrium*

Nash equilibrium is a set of strategies, one for each of the *n* players of a game, that has the property that each player's choice is his best response to the choices of the *n–1* other players. It would survive an announcement test: if all players announced their strategies simultaneously, nobody would want to reconsider.

Ex:

If two players Alice and Bob choose strategies A and B, (A, B) is a Nash equilibrium if Alice has no other strategy available that does better than A at maximizing her payoff in response to Bob choosing B, and Bob has no other strategy available that does better than B at maximizing his payoff in response to Alice choosing A. In a game in which Carol and Dan are also players, (A, B, C, D) is a Nash equilibrium if A is Alice's best response to (B, C, D), B is Bob's best response to (A, C, D), and so forth.

*https://www.pnas.org/content/101/12/3999



The Structure of GAN

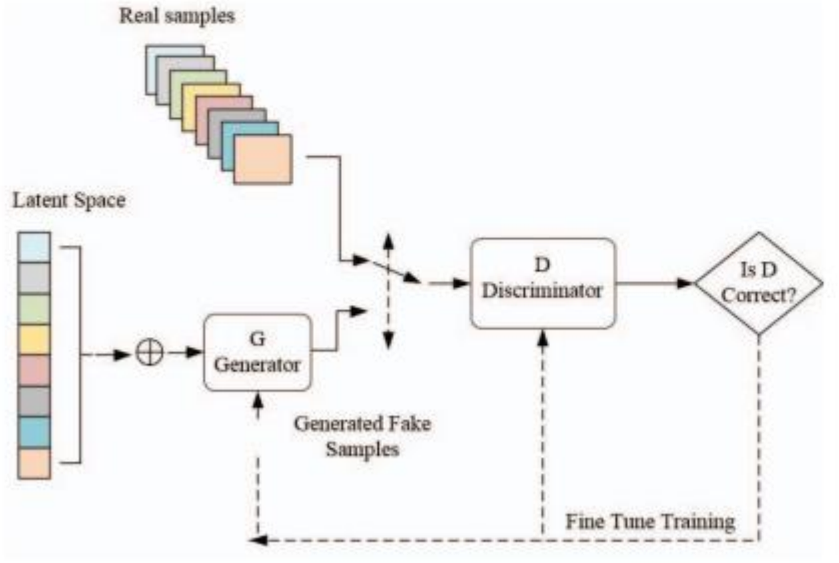
Any differentiable function can be used to represent the generator and discriminator of GANs, which means that the generator and discriminator can adopt the deep neural network.

differentiable function and NN

- differential equations are an important tool in describing the behavior of complex systems. Using differential equations models in our neural networks allows these models to be combined with neural networks approaches
- artificial neural networks (ANN) is to build up representations for complicated functions using compositions of relatively simple functions called layers.

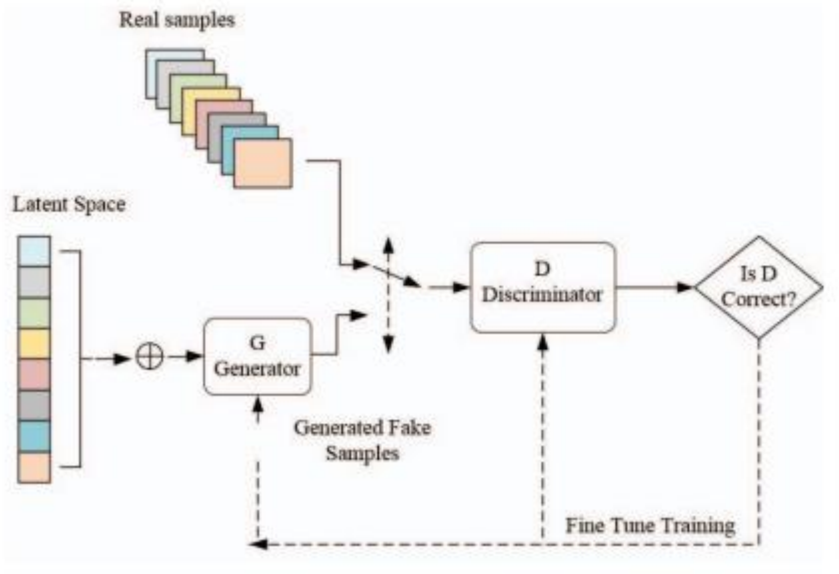
$$ANN(x) = L_n \circ L_{n-1} \circ \dots L_2 \circ L_1(x)$$

- If our layers are differentiable then we can find the gradient of this cost function and use this to find a local minimum of the cost in an efficient manner.



The Structure of GAN

If the differentiable functions D and G are used to represent discriminators and generators respectively, their inputs are real data x and random variable z respectively.



The Structure of GAN

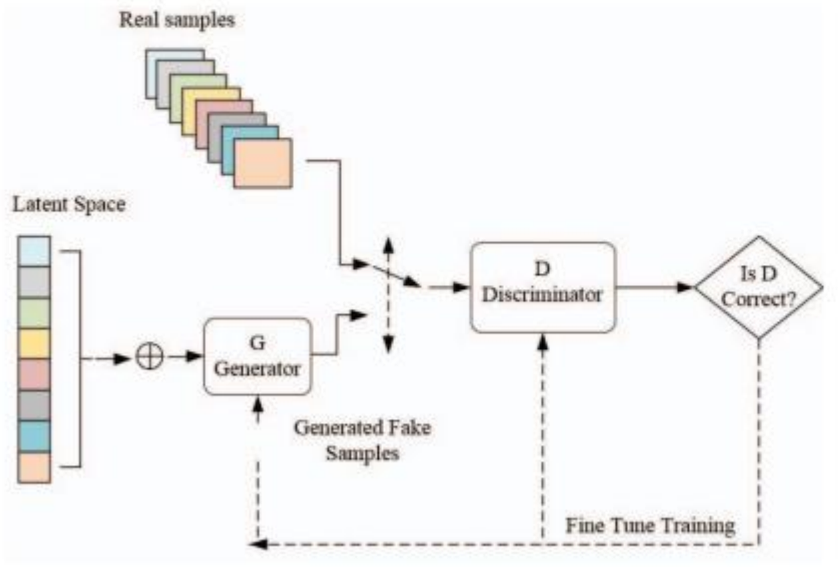
G(z) is a sample generated by G, which obeys the distribution of real data.

If the input of the discriminator comes from the real data, it is tagged as One.

If the input sample is G(z), it is tagged as Zero.

The goal of D is to achieve a correct binary classification of data sources: true(from the distribution classification of real data x) or false(from the fake data G(z) of the generator).

The goal of G is to make the self generated false data G(z) perform as the same as the real data x performing on D(x),



The Structure of GAN

The performance of D and G can be improved by the process and iteratively optimized.

When the discriminative ability of D is improved to a certain degree and the data source can not be correctly identified, the generator G can be considered to have learned the distribution of the actual data.

Architecture

Neural networks modelling essentially requires to define two things: an architecture and a loss function.

- a generative network $G(.)$ that takes a random input z with density p_z and returns an output $x_g = G(z)$ that should follow (after training) the targeted probability distribution
    - a discriminative network $D(.)$ that takes an input x that can be a "true" one (x_t, whose density is denoted p_t) or a "generated" one (x_g, whose density p_g is the density induced by the density p_z going through G) and that returns the probability $D(x)$ of x to be a "true" data

loss function of GANs

- If we send to the discriminator “true” and “generated” data in the same proportions, the expected absolute error of the discriminator can then be expressed as:

$$\begin{aligned} E(G,D) &= \frac{1}{2}\mathbb{E}_{x\sim p_t}[1-D(x)] + \frac{1}{2}\mathbb{E}_{z\sim p_z}[D(G(z))] \\ &= \frac{1}{2}\left(\mathbb{E}_{x\sim p_t}[1-D(x)] + \mathbb{E}_{x\sim p_g}[D(x)]\right) \end{aligned}$$

- The goal of the generator is to fool the discriminator whose goal is to be able to distinguish between true and generated data. So, when training the generator, we want to maximise this error while we try to minimise it for the discriminator.

$$\max_G \left(\min_D E(G,D)\right)$$

[Roughly, the expectation is **the average value of the random variable where each value is weighted according to its probability**.]

Advantage

- Compared to other generative models, higher quality samples (sharper and clearer images) can be produced by GAN than other models.
- Shorter runtime will consumed by GAN

Disadvantage

- the generator degeneration, continuously generate the same sample points generation, unable to continue learning.
- When the generative model collapses, the discriminative model also points to the same direction for similar sample points, and the training can not continue.
- Furthermore, although the samples generated by GANs are diverse, there exists the collapse mode problem.
- **Mode collapse refers to the scenarios in which the generator makes multiple images that contain the same color or texture themes, thereby having little difference for human understanding**

Practical Application*: Edge Computing Empowered Generative Adversarial Networks for Realtime Road Sensing

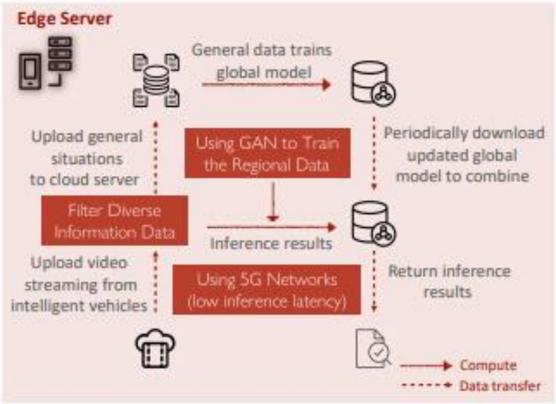
*https://people.ece.ubc.ca/xiaoyif/Papers/Generative-IWQoS.pdf

Realtime Road Sensing

To cope with the huge network traffic and high computational demands, as well as to improve the system response time and dynamically tune the trained models, apply the concept of edge computing to the domain of intelligent road sensing services.

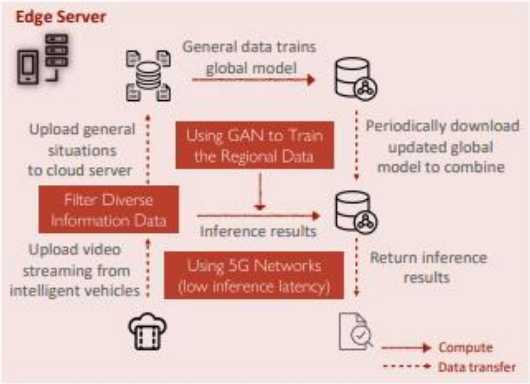
The wide deployment of 5G network can bring many benefits, such as low inference latency, online updating models and maintenance.

Its extending services on the edge servers can tune the general road sensing model with its own specific local data circumstances.



Challenge

- dataset for specific local cases is often not large enough, causing that the dataset may not contain enough diverse information
- performance of most deep learning technologies highly depend on largescale training data.
- The training datasets collected from one regional area may not be enough.
- the edge server needs to spend a long time to collect enough data to improve the performance of the model.



Solution:GAN Based Data Augmentation

- GAN to improve the performance of the road sensing model running on the edge:
    - Instead of collecting all the possible situations, **GAN generate examples that will be challenging for the model to classify**, so that the road sensing model can become more robust to different specific local conditions.
- generate hard examples for an object detector to classify **without increasing the cost for collecting large-scale training data** under various specific local conditions.

(a) The Fast-RCNN model can detect road signs under good illumination condition

(b) The stop sign recognition with reflection using the discriminator with GAN on the edge computing

Fig. The examples of traffic sign recognition illustrate the enhancement of road sensing on the edge in special conditions