Vidya Vikas Education Trust's

Universal College of Engineering

Approved by AICTE, DTE, Maharashtra State Government and Affiliated to University of Mumbai

Accredited with 'B+' grade by NAAC | Recognised as Linguistic (Gujarati) Minority Institution

Vol 03 Edition 6
Dec 2020

Coffee & Code ;

An Initiative by the Department of Computer Engineering

VISION

To be recognized as a department that provides quality technical education and research opportunities that eventually caters to helping and serving the community.

MISSION

- To groom the students to participate in curricular and co-curricular activities by providing efficient resources.
- To motivate the students to solve real world problems to help the society grow.
- To provide a learning ambience to enhance innovations, team spirit and leadership qualities for students.

Contents:

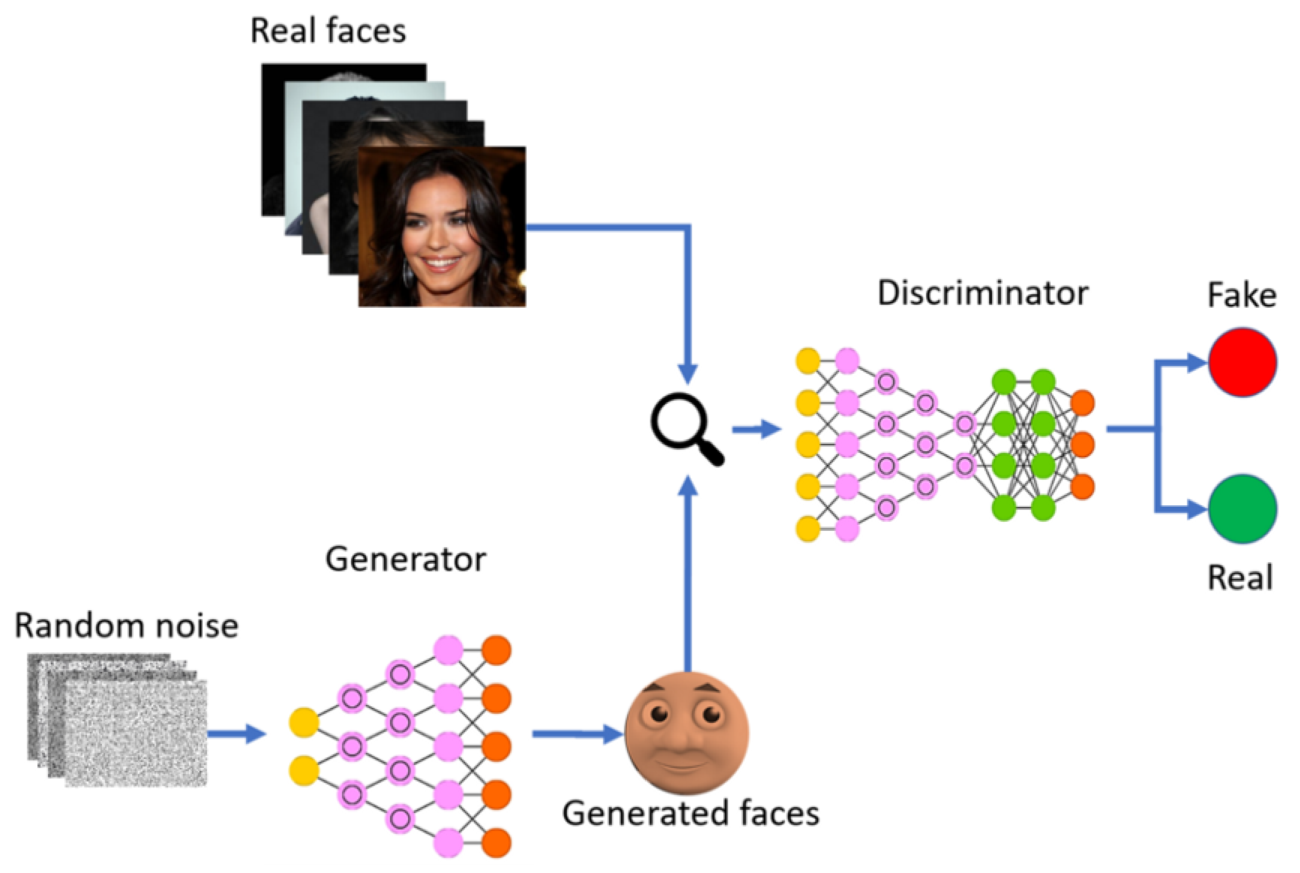
Prepared by:

MS. APURVA CHAUDHARI
MRS. HEZAL LOPES
MR. SRIDHAR IYER

In Association with
CESA
(Computer Engineering Student Association)

A Brief Introduction To GANs

GANs, or Generative Adversarial Networks, are a type of neural network architecture that allows neural networks to generate data. In the past few years, they've become one of the hottest subfields in deep learning, going from generating fuzzy images of digits to photorealistic images of faces.



Variants of GANs have now done insane stuff, like converting images of zebras to horses and vice versa.

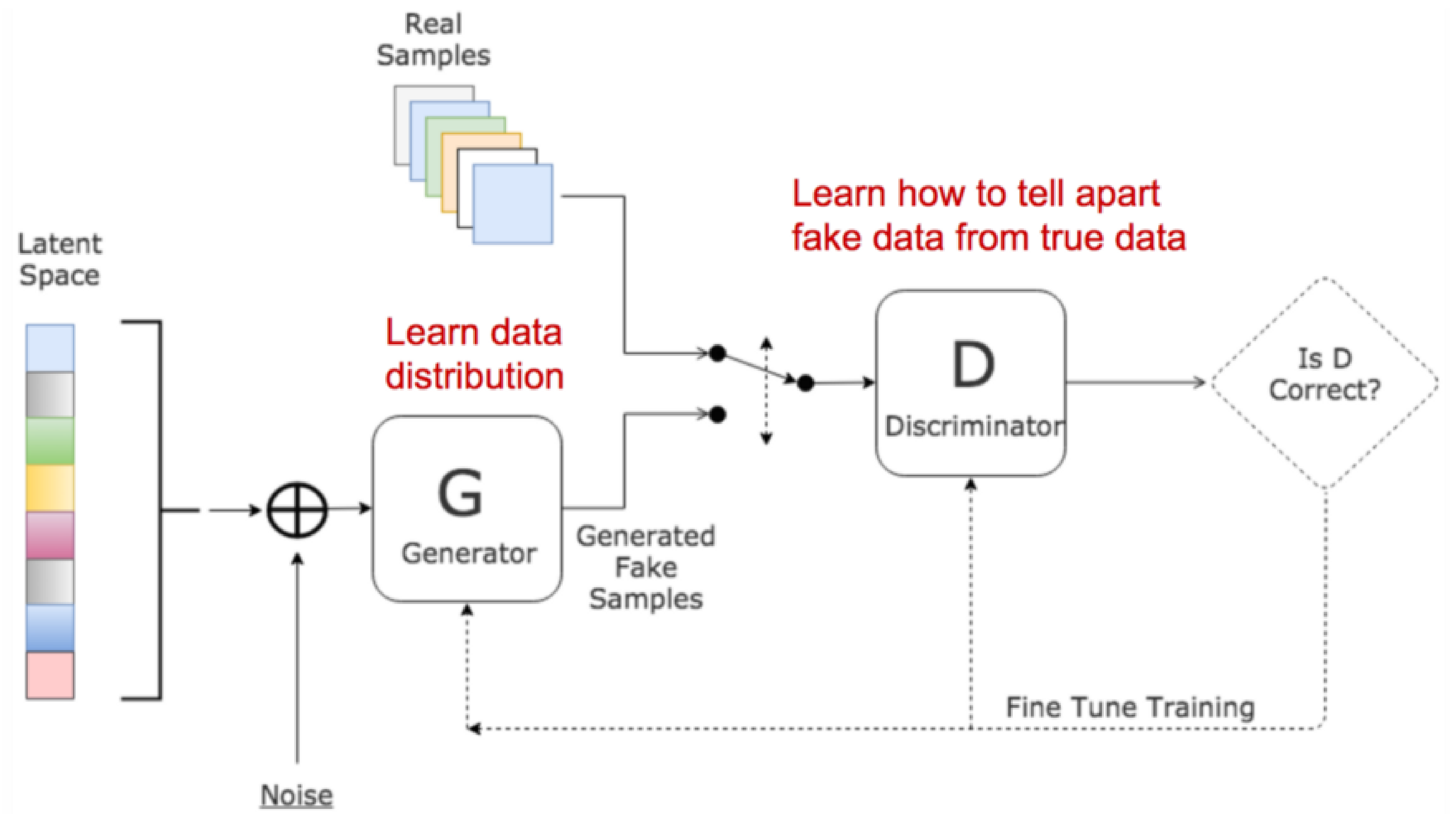
So how do GANs work?

GANs learn a probability distribution of a dataset by pitting two neural networks against each other. Here's a great article that explains probability distributions and other concepts for those who aren't familiar with them:

https://towardsdatascience.com/probability-concepts-explained-probability-distributions-introduction-part-3-4a5db81858dc

One model, the generator, acts akin to a painting forger. It tries to create images that look very similar to the dataset. The other model, the discriminator, acts as the police and tries to detect whether the images generated were fake or not.

What basically happens, is that the forger keeps getting better at making fakes, while the police keep getting better at detecting fakes. Effectively, these two models keep trying to beat each other, until, after many iterations, the generator creates images indistinguishable from the real dataset. neural networks against each other.



Training generative adversarial networks involve two objectives:

1. The discriminator maximizes the probability of assigning the correct label to both training examples and images generated by the generator. I.e the policeman becomes better at differentiating between fakes and real paintings.

2.The generator minimizes the probability that the discriminator can predict that what it generates is fake. I.e the generator becomes better at creating fakes

Let’s try and encode these two ideas into a program. We’ll be following this code in this tutorial.

https://github.com/sarvasvkulpati/intro_to_gans

The Data:

GANs need a dataset to use, so for this tutorial, we’ll be using the classic hello world to machine learning — MNIST, a dataset of handwritten digits. The generator also needs random input vectors to generate images, and for this, we’ll be using numpy.

The GAN Function:

The GAN plays a minimax game, where the entire network attempts to optimize the function V(D,G). This is the equation that defines what a GAN is doing:

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

Now to anyone who isn’t well versed in the math behind it, it looks terrifying, but the idea it represents is simple, yet powerful. It’s just a mathematical representation of the two objectives as defined above.

The generator is defined by G(z), which converts some noise z we input into some data, like images. The discriminator is defined by D(x), which outputs the probability that the input x came from the real dataset or not.

We want the predictions on the dataset by the discriminator to be as close to 1 as possible, and on the generator to be as close to 0 as possible. To achieve this, we use the log-likelihood of D(x) and 1-D(z) in the objective function. The log just makes sure that the closer it is to an incorrect value, the more it is penalized.

The generator is just a vanilla neural network model that takes a random input vector and outputs a 784-dim vector, which, when reshaped, becomes a 28* 28-pixel image.

The discriminator is another neural network that takes the output of the previous network, a 784-dimensional vector, and outputs a probability between 0 and 1 that it came from the training dataset.

Training the GAN:

1. **First, we load the data and split the data into several batches to feed into our model.**
2. **Here we just initialize our GAN network based on the methods defined above.**
3. **This is our training loop, where we run for the specified number of epochs.**
4. **We generate some random noise and take out some images from our dataset.**
5. **We generate some images using the generator and create a vector X that has some fake images and some real images.**
6. **We create a vector Y which has the “correct answers” that corresponds to X, with the fake images labeled 0 and the real images labeled 0.9.**
7. **They’re labeled 0.9 instead of 1 because it helps the GAN train better, a method called one-sided label smoothing.**
8. **We need to alternate the training between the discriminator and generator, so over here, we update the discriminatorFinally, we update the discriminator.**

SOURCE: https://medium.com/sigmoid/a-brief-introduction-to-gans-and-how-to-code-them-2620ee465c30

How 7 Lines of Code Turned Into a $36 Billion Empire

Two Irish brothers in their 20s outplayed the finance industry with seven lines of code

On June 24 last year, Patrick Collison, the co-founder of Stripe, posted a Tweet saying

"Hit our engagement metrics this weekend!"

Instead of chasing 1000-hour programming contracts to build clunky payments solutions for each individual client, the Collison brothers built 7 lines of code that developers could simply plug into their websites.

The result is Stripe — a company that has more cash than it knows what to do with. Here are the two main reasons why this worked so well.

1. They empowered developers.

When Stripe was launched in 2010, it was essentially a community of developers.

See, back in 2010, setting up a payments system for your business was an extremely tedious process. Most integrations at the time took weeks and months to set up. It was an exhaustive process, involving several departments — from legal to accounting.

In most companies, financial integrations were a top-down decision that came from managers who knew little about the technical nightmare of the integration process. They would set arbitrary goals and deadlines and pass the responsibility onto developers who had to outdo themselves in order to meet expectations.

Stripe changed all that by giving developers a quick and reliable way to set up payments. As a result, they built an army of champions for their product, effectively bypassing the “decision-makers,” most of whom would have never believed that 7 lines of code can replace custom integrations that took up to 6 months to build.

2. They made payments ridiculously simple.

Banks. Credit card companies. Regulators. Yes, the payments industry has a lot of police officers. And you need to get approval from all of them before the dollars falling into your bank account are considered legal.

Stripe changed all that. Collison brothers spent 2 years testing their platform and building relationships with credit card companies, banks, and regulatory institutions. John and Patrick spent countless hours presenting their product to authorities and building enough trust to partner with them. As a result, users only have to upload a couple of documents to set up their fully operational Stripe account.

Another reason why Stripe blew up is that it was so universal. Platforms like Shopify and Lyft rely on payment simplicity — each additional step in account verification can mean millions of churned users. Stripe’s 7 lines of core code worked just as well in any environment, enabling billions of emerging platform users with same-week verification.

Stripe = Genius + Humility

“They have the advantage of coming to California without being tainted and polluted by what’s in the water supply and air of Silicon Valley,” says Moritz, a partner at Sequoia Capital and a Stripe board member. “They’re more humble and well-rounded.”

Apparently, then, the good guys do win from time to time.

SOURCE: https://entrepreneurshandbook.co/two-reasons-why-these-7-lines-of-code-turned-into-a-36-billion-empire-6d2b2d1a8da2

Building a budget PC for Deep Learning 2020

You might have already seen a lot of people using dual 2080Ti or TITAN for running their Deep Learning algorithms. Sorry but that's very expensive for us. We will be settling down with RTX 2060, 1070 Ti or 1660 Super.

The best thing about building a custom PC is, that you can swap your graphics card anytime if you want to update it later.

— — — — Starting with GPU — — — — -

Let's choose RTX 2060 for this Squad

(Remember if you get GTX 1070 Ti for lower prices always go for that)

— — — — — — — — CPU — — — — — — — — -

Ryzen 5 3600X

But does the AMD CPU runs Deep learning algorithms? Deep Learning algorithms always runs on CUDA cores, it is the Machine Learning algorithm that runs in your processor or some light. Even if you get Intel CPU that's completely fine.

— — — — — — — — — RAM — — — — — — — — —

16 GB of RAM is enough for these SQUAD
(if budget is enough you can always extend)

— — —So what about Cooling and PSU?— — —

Since water cooling will not fit into our budget but many of the Air Cooler will make our done. There are a lot of air coolers for the CPU. As I already said, we are only focusing on our GPU.
And PSU, 650W, 600W for this budget segment. You can choose anything between this. Don't get confused with Gold, Bronze, or Silver Edition. Even with the Bronze edition, your PC will run perfectly.

— — — — — — — —Storage — — — — — — — —

For this budget segment, you can easily get a SSD and Hard drive.
And, the fact is since you are going to store a lot of datasets for Deep Learning to be trained, for a beginner or learning purpose 1 TB is enough for you, since you can upgrade it later anytime

Source : https://medium.com/datadriveninvestor/building-a-budget-pc-for-deep-learning-2020-f9a2e07a4aaf

ALUMNI MEET

When: 28th Nov, 2020, 5:00pm to 7:00pm

Alumni meets are always a worthy occasion for the passed out students, where they can take a pause and remember the golden past. Alumni Meet is an experience for both the institute and its alumni. An alumni meet was organized on 28th November 2020 on Google Meet from 5 pm-7 pm.

Campus Director, Dr. J. B. Patil had welcomed all the alumni's with a wonderful memory lane of their college days. Students were feeling nostalgic to be with their teachers, mentors, friends, and colleagues. All the existing office bearers of the Alumni Association and other alumni have participated along with faculty members from all departments.

The event was stimulating and enjoyable and simultaneously profitable as all members shared their views, new ideas, information, and insights came up. Also, many fun games were organized, which all alumni's had a lot of fun.

Scan Me for our previous Editions

You can send your articles to the following email ids:

sridhar.iyer@universal.edu.in, hezal.lopes@universal.edu.in , apurva.chaudhari@universal.edu.in