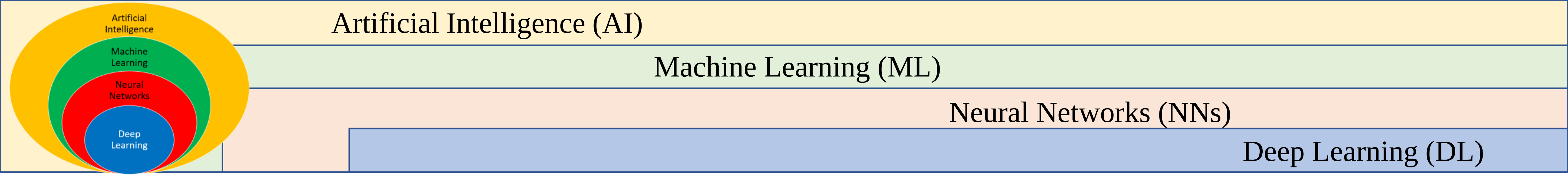
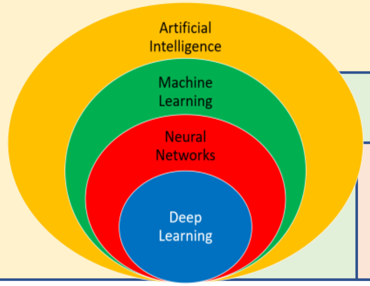


Advanced Artificial Intelligence

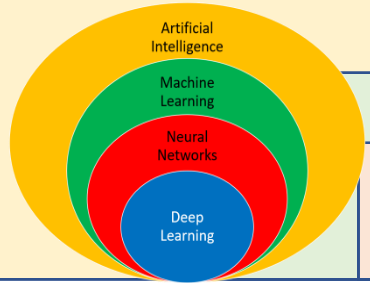
Dr. Rastgoo

2022


Generative models



Part 2: Generative Adversarial Network (GAN)



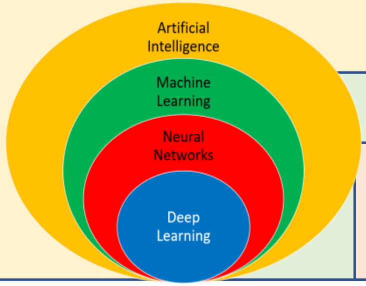
Introduction

- GANs introduce the concept of adversarial learning, as they lie in the rivalry between two neural networks.
- These techniques have enabled researchers to create realistic-looking but entirely computer generated photos of people’s faces.
- They have also allowed the creation of controversial “deepfake” videos.
- Actually, GANs can be used to imitate any data distribution (image, text, sound, etc.).

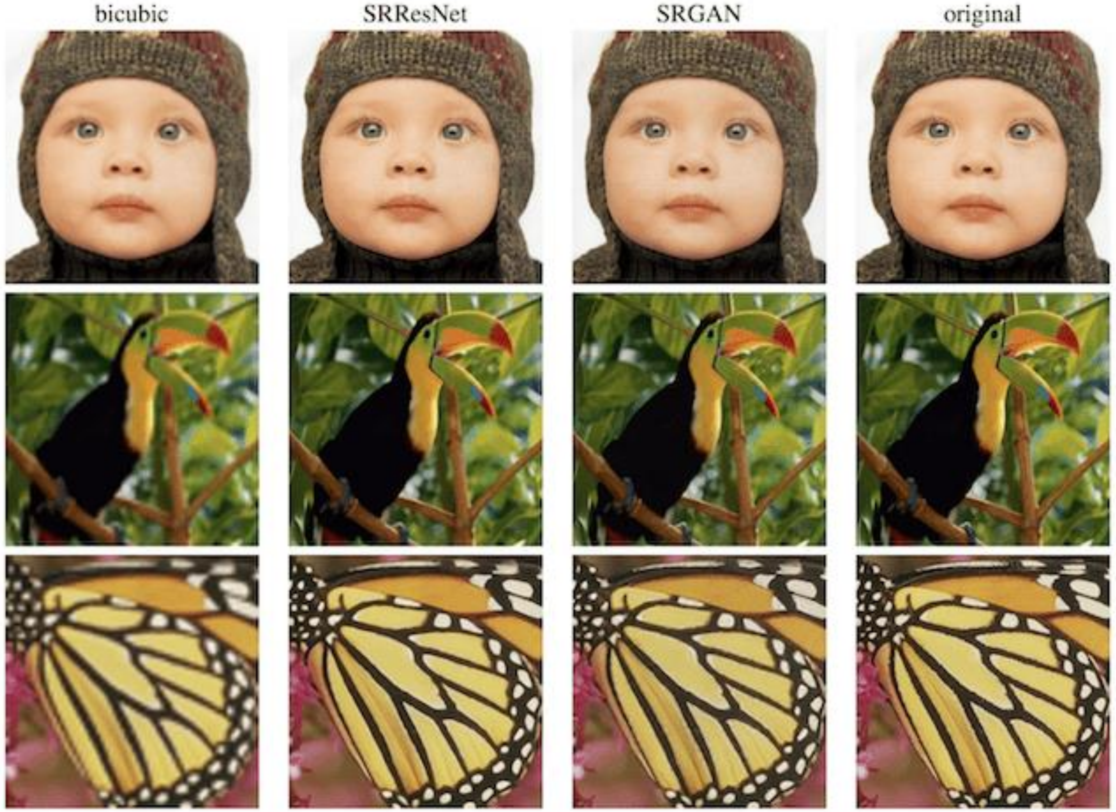
Introduction

- An example of GANs' results from 2018 is given in below figure (Figure 1).
- These images are fake yet very realistic.
- The generation of these fictional celebrity portraits, from the database of real portraits Celeba-HQ composed of 30,000 images, took 19 days. The generated images have a size of 1024×1024.

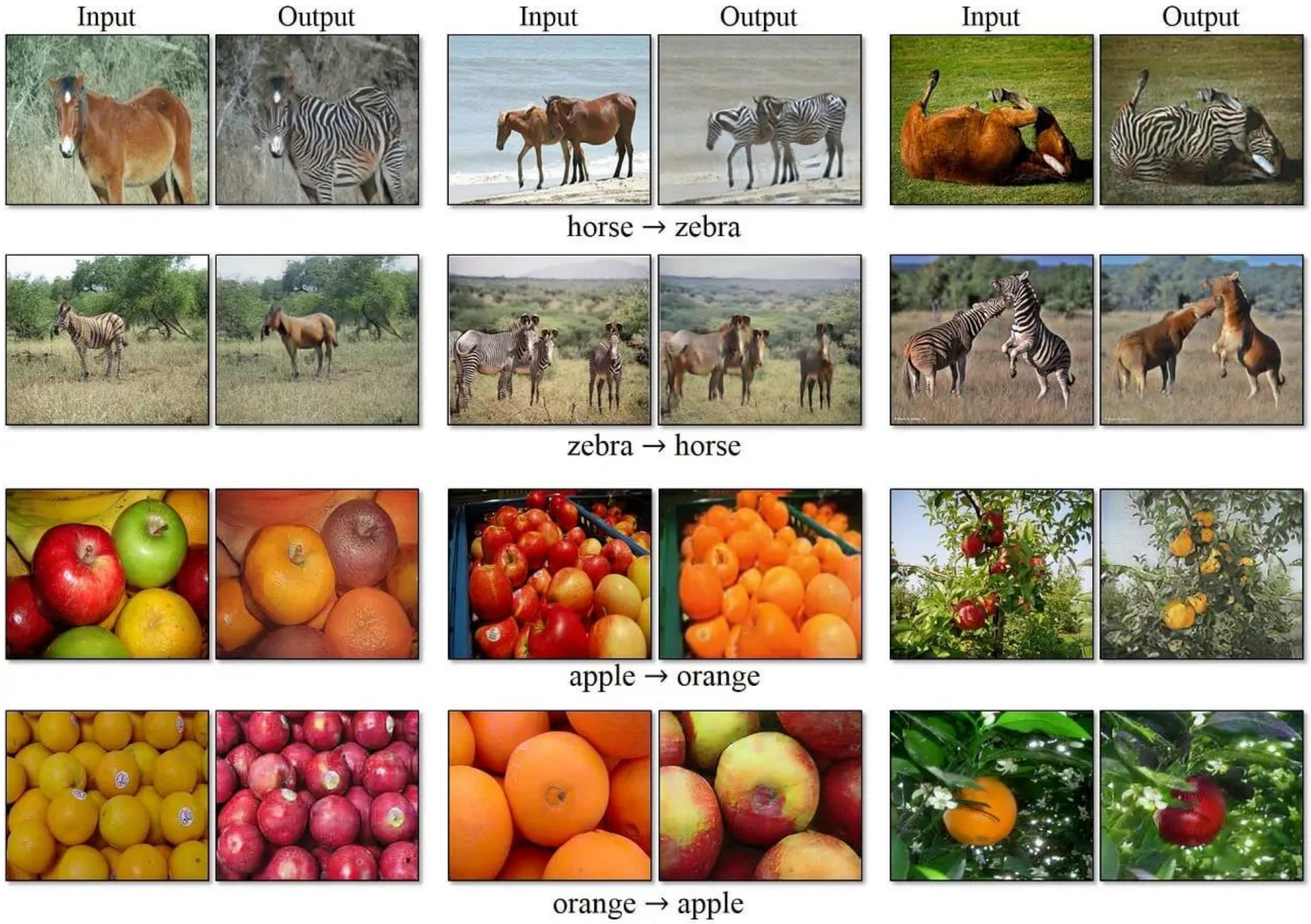
Figure 1



Introduction

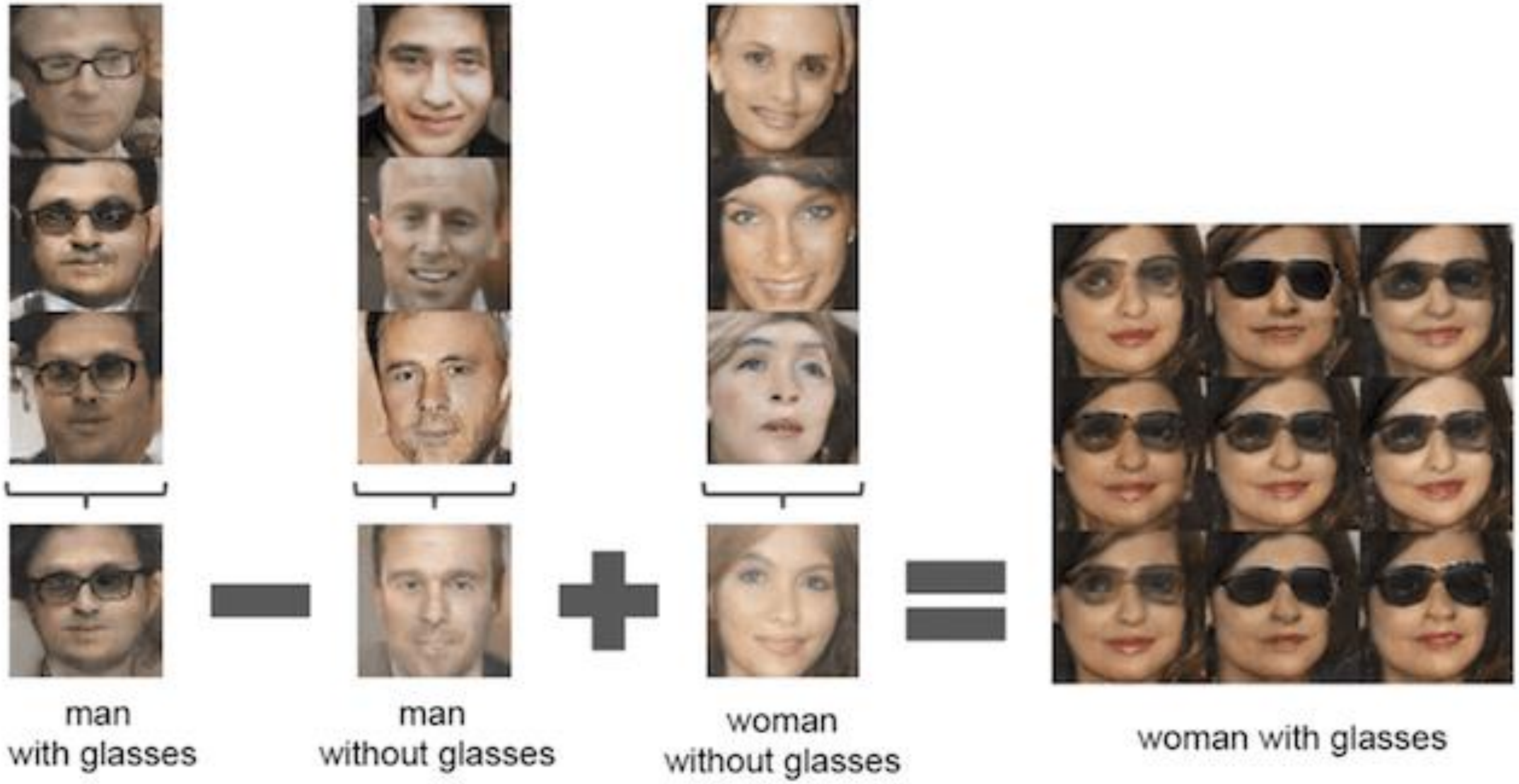


Introduction



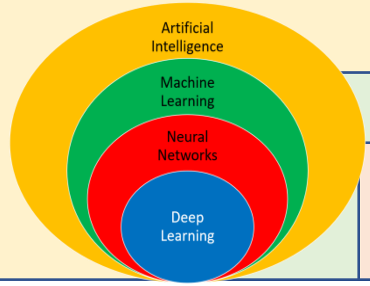
Introduction

Introduction

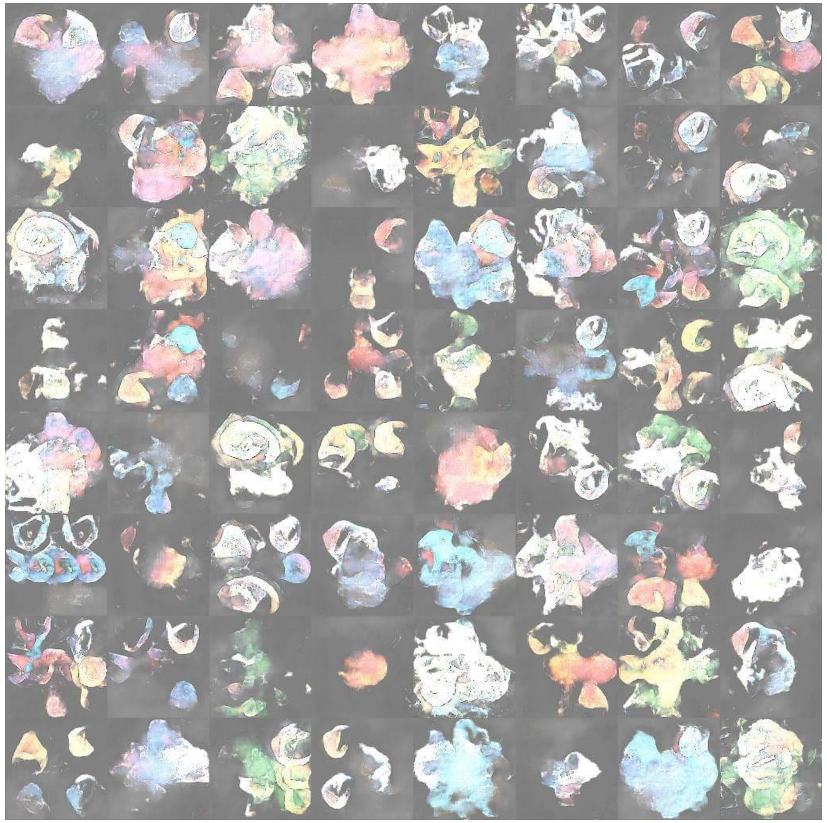


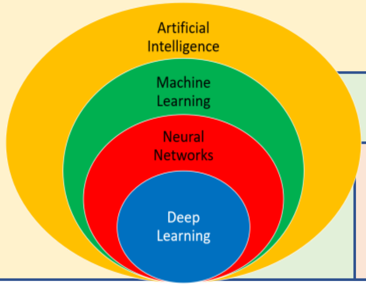
Introduction

Introduction

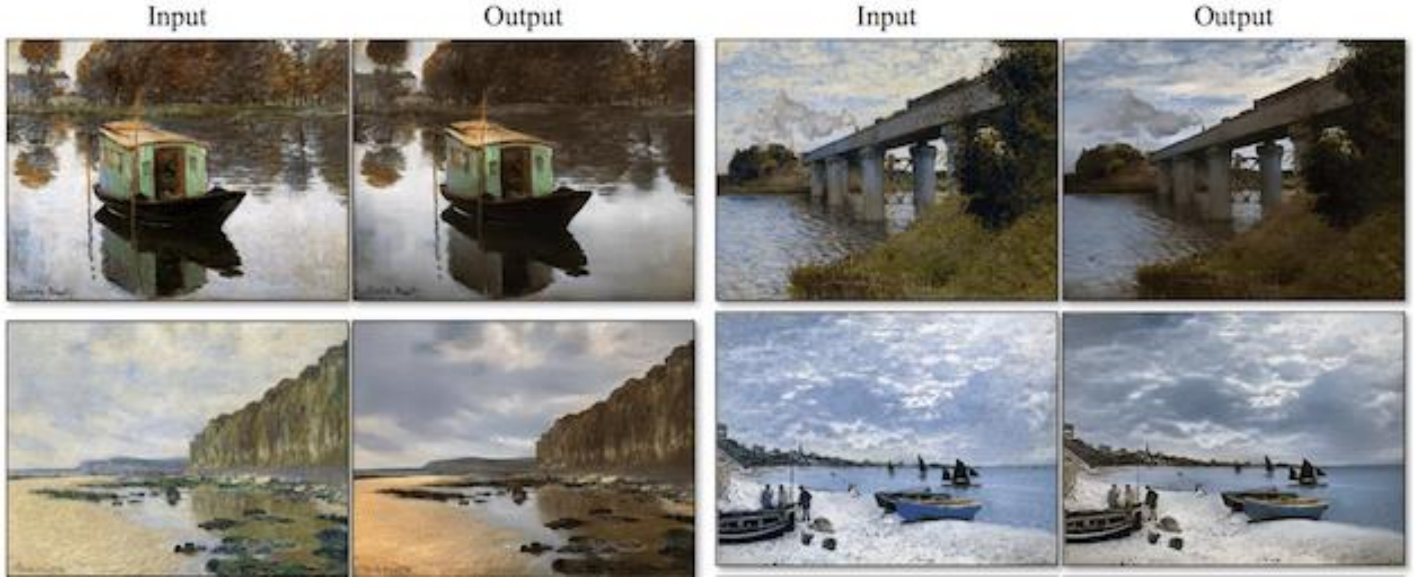


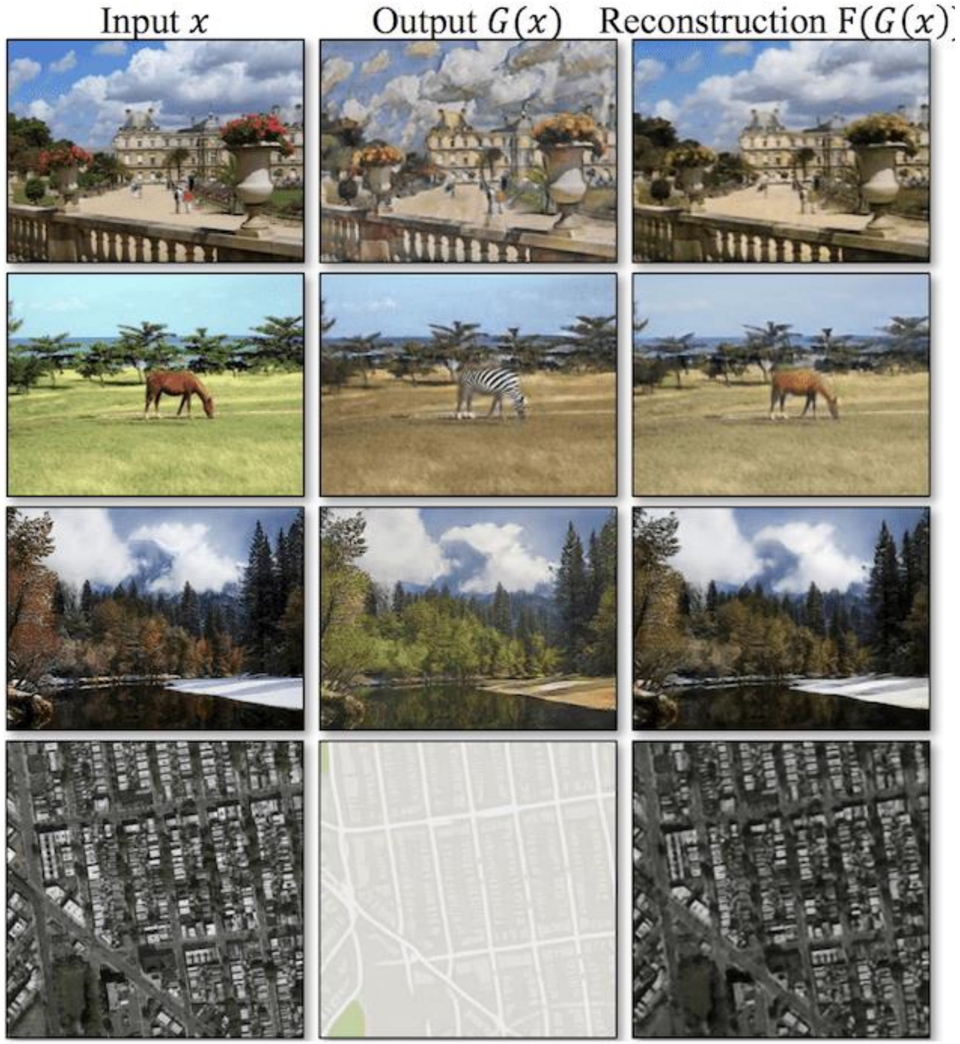
Introduction



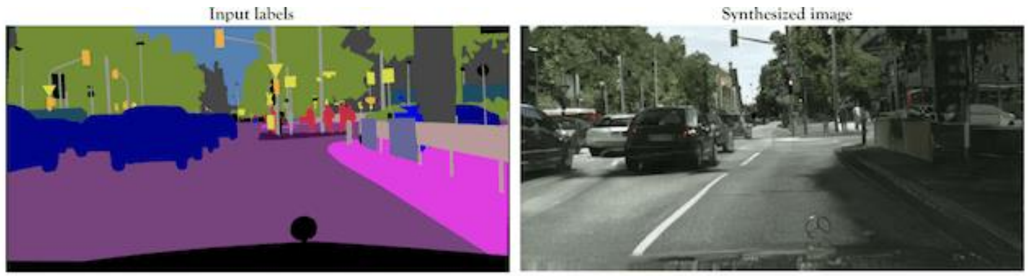


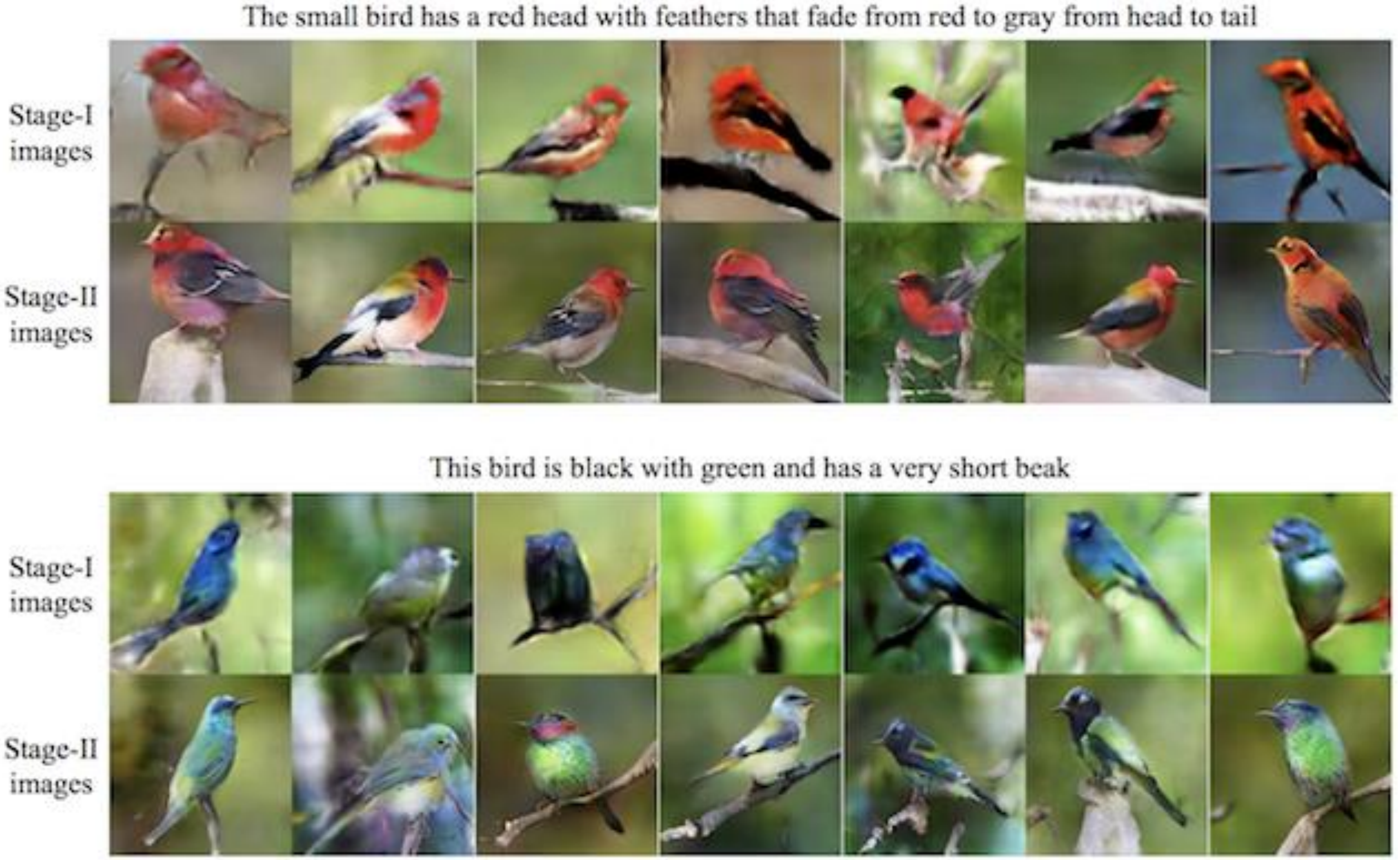
Introduction

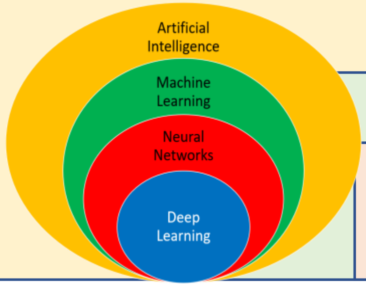




Introduction



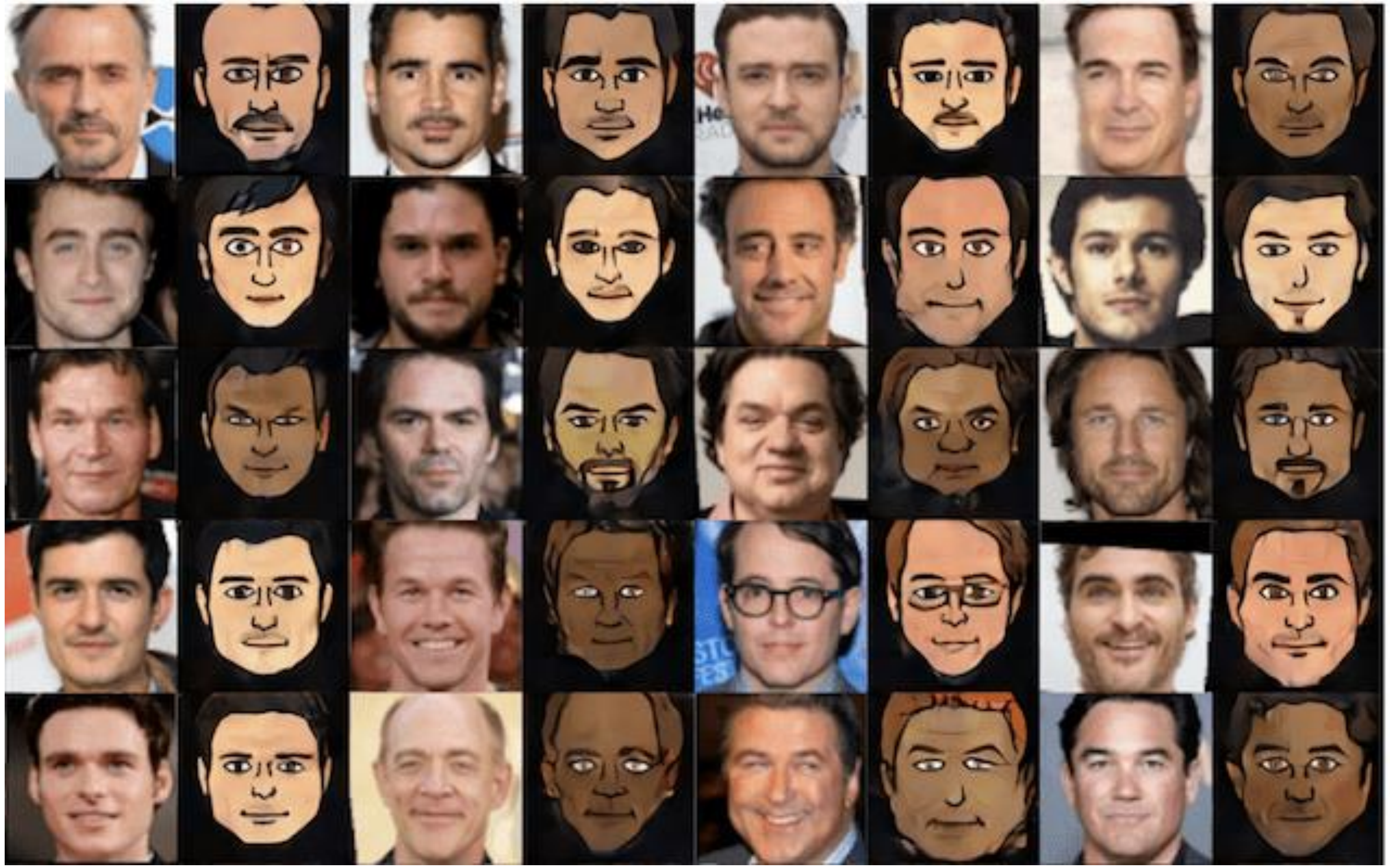




Introduction

(a) (b)

(c) (d)



How do GANs work?

- Generative adversarial networks (GANs) are a generative model with implicit density estimation, part of unsupervised learning and are using two neural networks.

- Thus, we understand the terms “generative” and “networks” in “generative adversarial networks”.

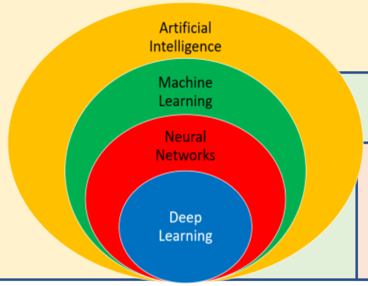
The principle: generator vs discriminator

- The principle is a two-player game: a neural network called the generator and a neural network called the discriminator.

- The generator tries to fool the discriminator by generating real-looking images while the discriminator tries to distinguish between real and fake images.

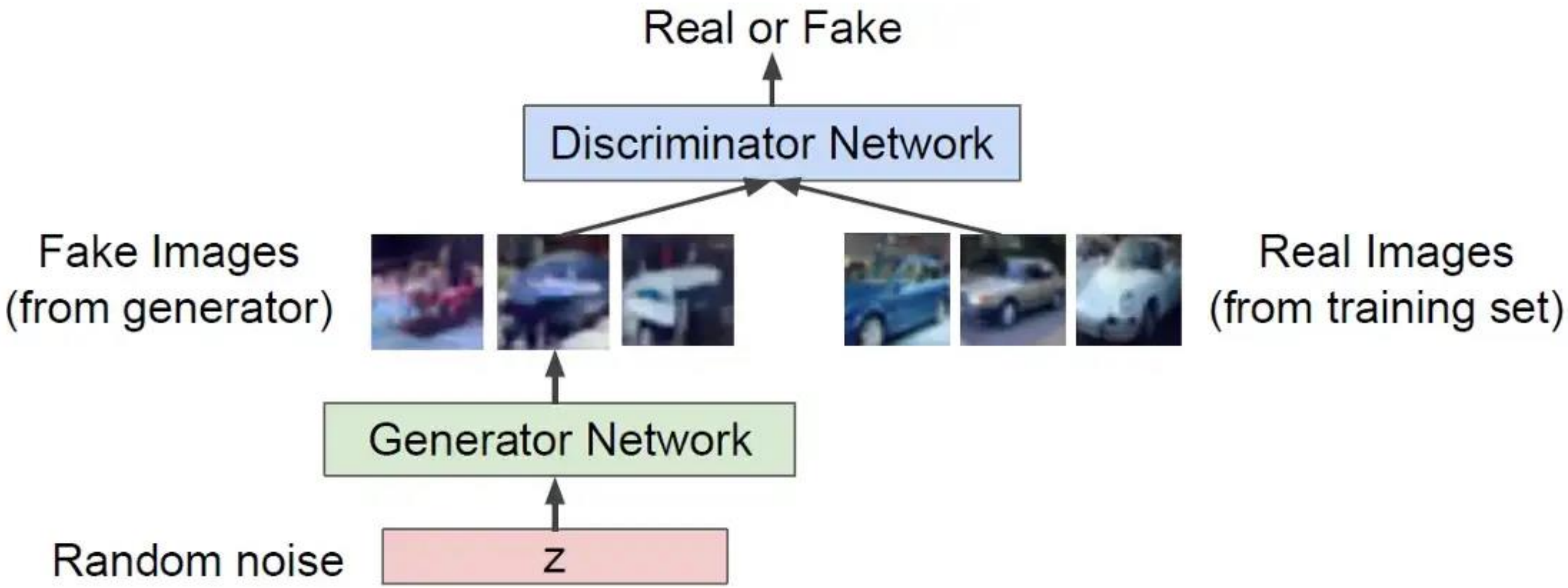
- At the bottom left of Figure 2, we can see that our generator samples from a simple distribution: random noise.

- The generator can be interpreted as an artist and the discriminator as an art critic. (See Figure 3)



The principle: generator vs discriminator

Figure 2



The principle: generator vs discriminator

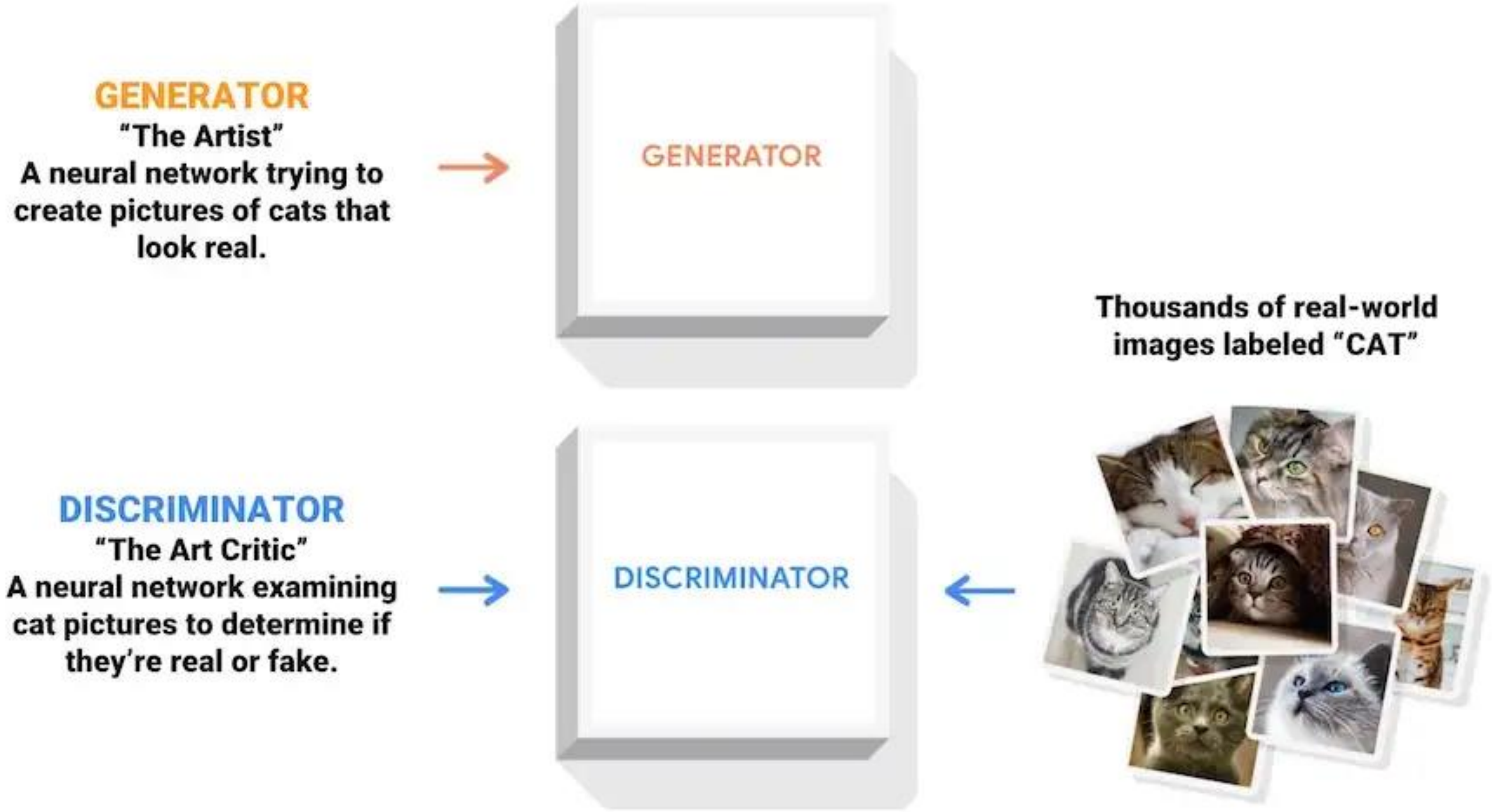


Figure 3

The principle: generator vs discriminator

- During training, the generator progressively becomes better at creating images that look real, while the discriminator becomes better at telling them apart.

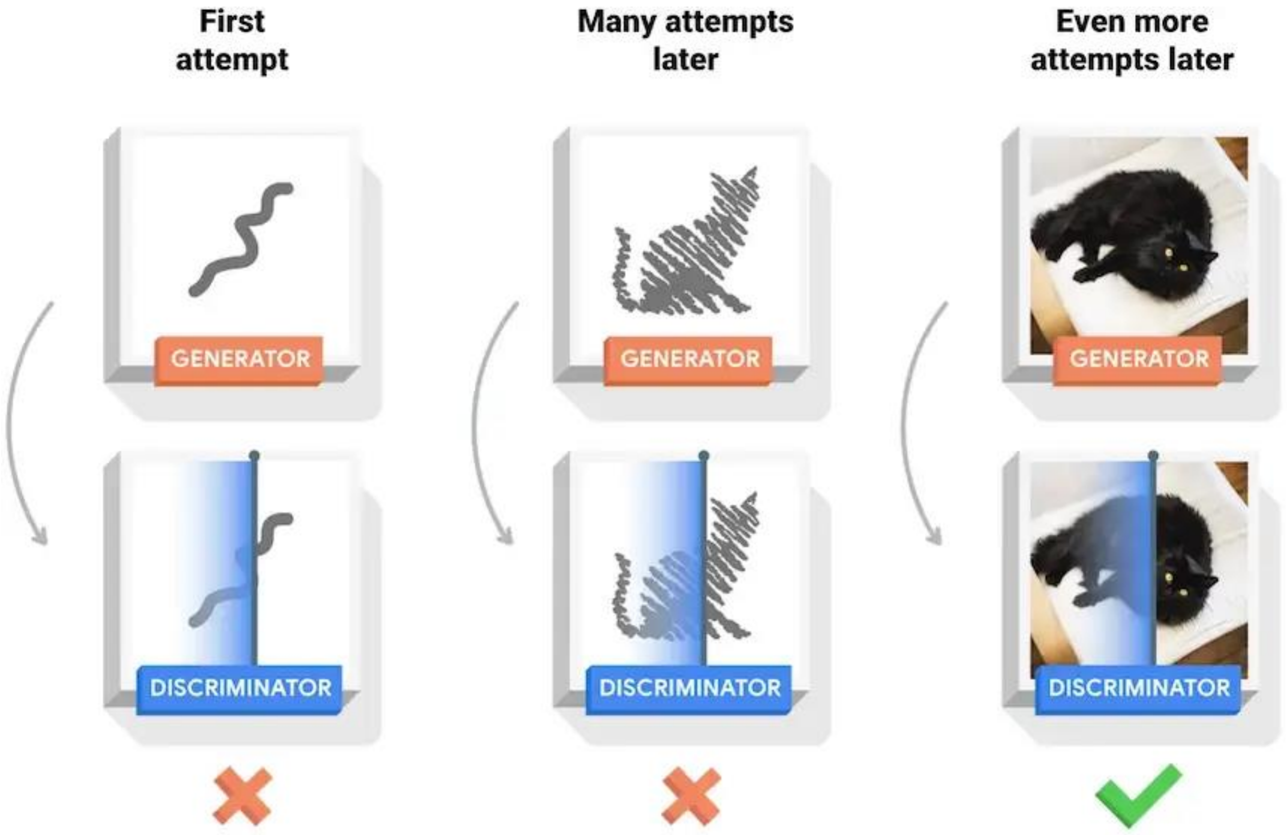
- The process reaches equilibrium when the discriminator can no longer distinguish real from fake images. See Figure 4.

- Thus, if the discriminator is well trained and the generator manages to generate real-looking images that fool the discriminator, then we have a good generative model:

We are generating images that look like the training set!

The principle: generator vs discriminator

Figure 4



The principle: generator vs discriminator

- After this training phase, we only need the generator to sample new (false) realistic data.
- We no longer need the discriminator.
- Note that the random noise guarantees that the generator does not always produce the same image (which can fool the discriminator).
- Note that at the beginning of the training, the generator only generates a random noise that does not look like the training data.

Mathematically: The two-player minimax game

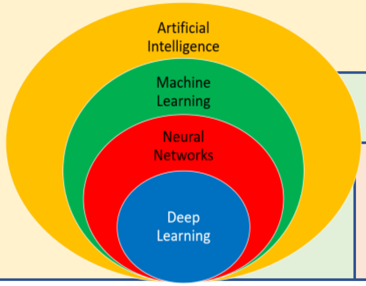
- The generator G and the discriminator D are jointly trained in a two-player minimax game formulation.

- The minimax objective function is:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{\text{data}}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log \left(1 - D_{\theta_d} \left(G_{\theta_g}(z) \right) \right) \right]$$

parameters of *G*

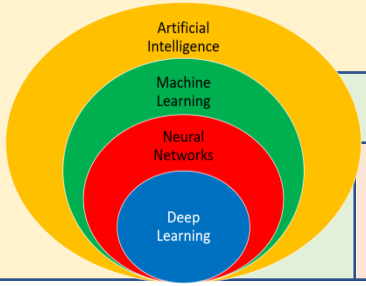
parameters of *D*



Mathematically: The two-player minimax game

- By definition, D outputs the likelihood of real image in interval [0, 1]:
- • D(x) equals 1 (or is close to 1) if D considers that x is a real data,
- • D(x) equals 0 (or is close to 0) if D considers that x is a fake data (e.g. a generated data).

- We can prove that, at the equilibrium, D outputs 1/2 everywhere because D has no idea how to distinguish fake generated data from real data.

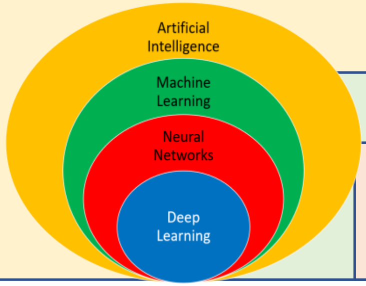


Mathematically: The two-player minimax game

- Because $x \sim p_{data}$, x is a real data.
- By definition of G, G(z) is a fake generated data.
- For example, x would be a real-life image of a cat and G(z) would be a fake generated image of a cat.
- Thus, D(x) is the output of the discriminator for a real input x and D(G(z)) is the output of the discriminator for a fake generated data G(z).

Mathematically: The two-player minimax game

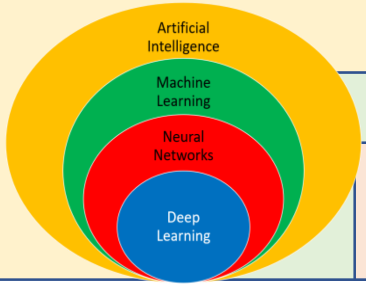
- The two-player minimax game from Equation (1) was written such that:
- The discriminator D tries to distinguish between real data x and fake data G(z).
- More precisely, the discriminator D plays with θ_d (θ_g being fixed) to maximize the objective function such that D(x) is close to 1 (x being real data) and such that D(G(z)) is close to 0 (a generated data is detected as false).



Mathematically: The two-player minimax game

- The generator G tries to fool the discriminator D into thinking that its fake generated data is real.

- More precisely, the generator G plays with θ_g (θ_d being fixed) to minimize the objective function such that D(G(z)) is close to 1 (a false generated data is detected as true by the discriminator).



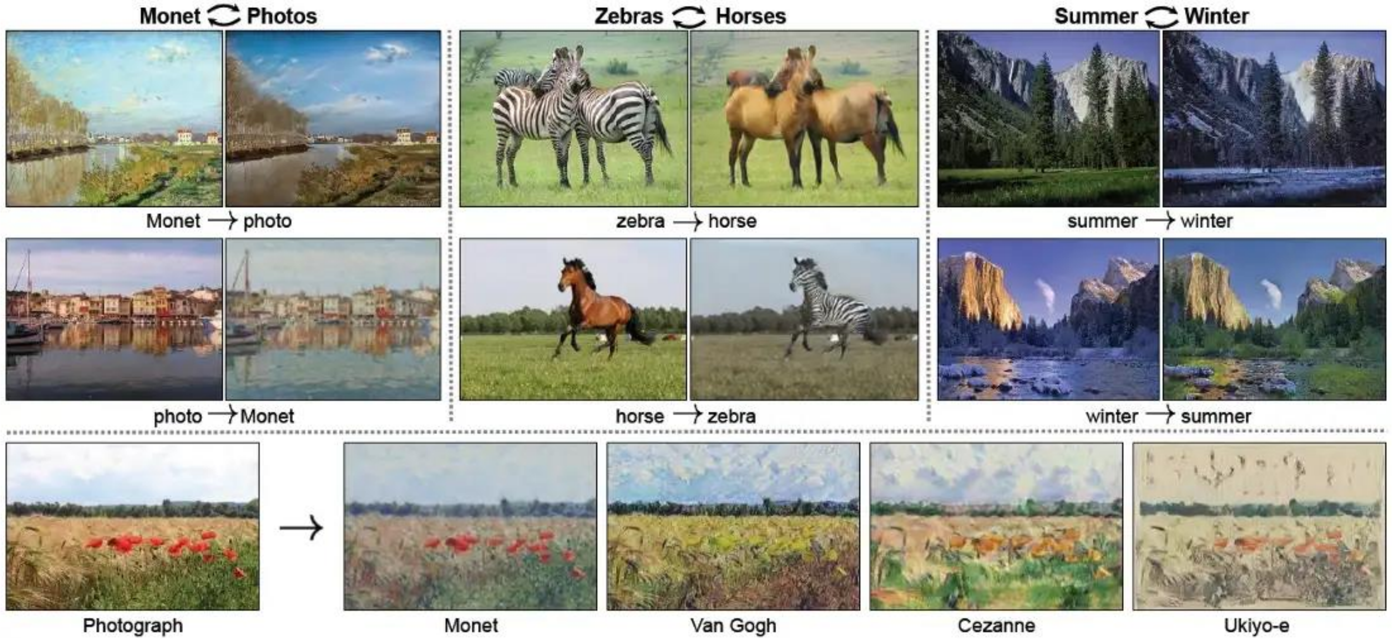
Mathematically: The two-player minimax game

- Although we are in unsupervised learning (the data is not labeled), we choose that the data generated by G has a 0 label for false (regardless of what the discriminator returns) and the real learning data has a 1 label for true. We can thus define a loss function.

Why are GANs so interesting?

- Generative models have several very useful applications: colorization, super-resolution, generation of artworks, etc. In general, the advantage of using a simulated model over the real model is that the computation can be faster.

Figure 5



Why are GANs so interesting?

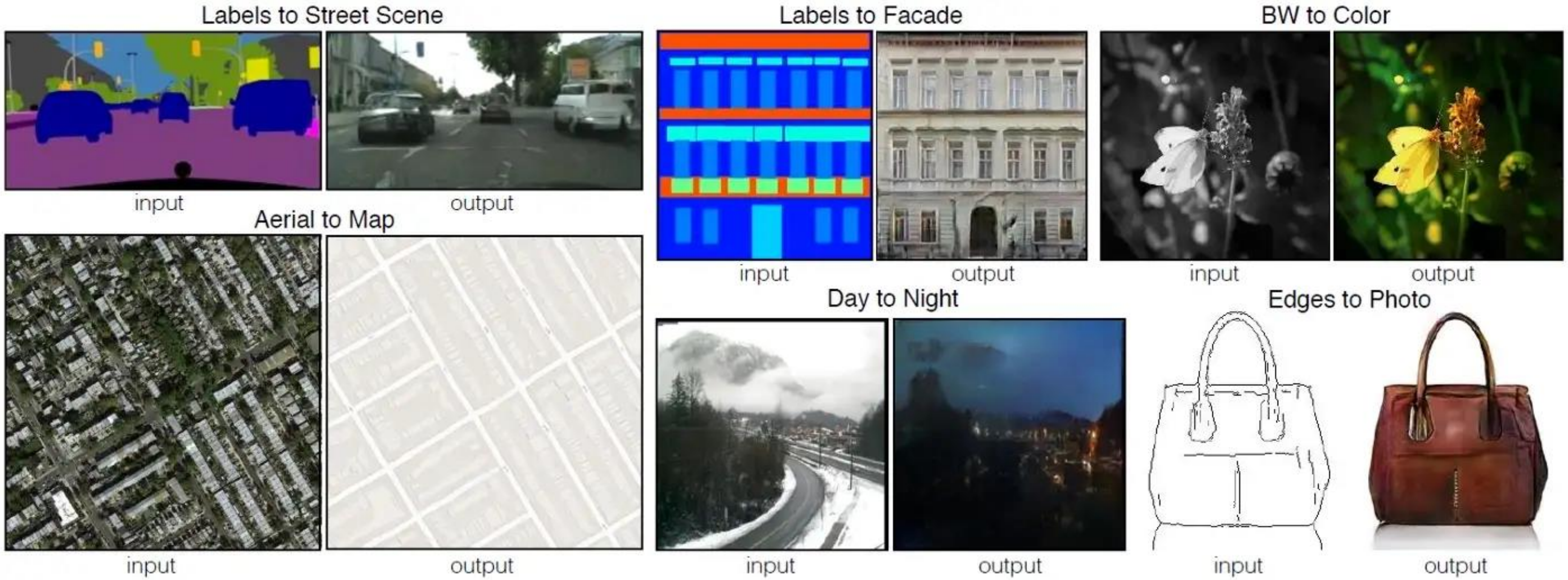
- One example is given Figure 5.

- These real images are transposed into realistic fictional images – or vice versa – with the CycleGAN developed by researchers at the University of Berkeley.

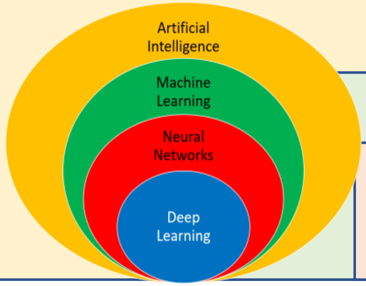
- The concept, called image-to-image translation, is a class of vision and graphics problems where the goal is to learn the mapping between an input image and an output image using a training set of aligned image pairs.

Why are GANs so interesting?

Figure 6



edges to photo



GAN Problems

- Many GAN models suffer the following major problems:

- Non-convergence: The model parameters oscillate, destabilize and never converge,
- Mode collapse: The generator collapses which produces limited varieties of samples,
- Diminished gradient: The discriminator gets too successful that the generator gradient vanishes and learns nothing,
- Unbalance between the generator and discriminator causing overfitting,
- Highly sensitive to the hyperparameter selections.

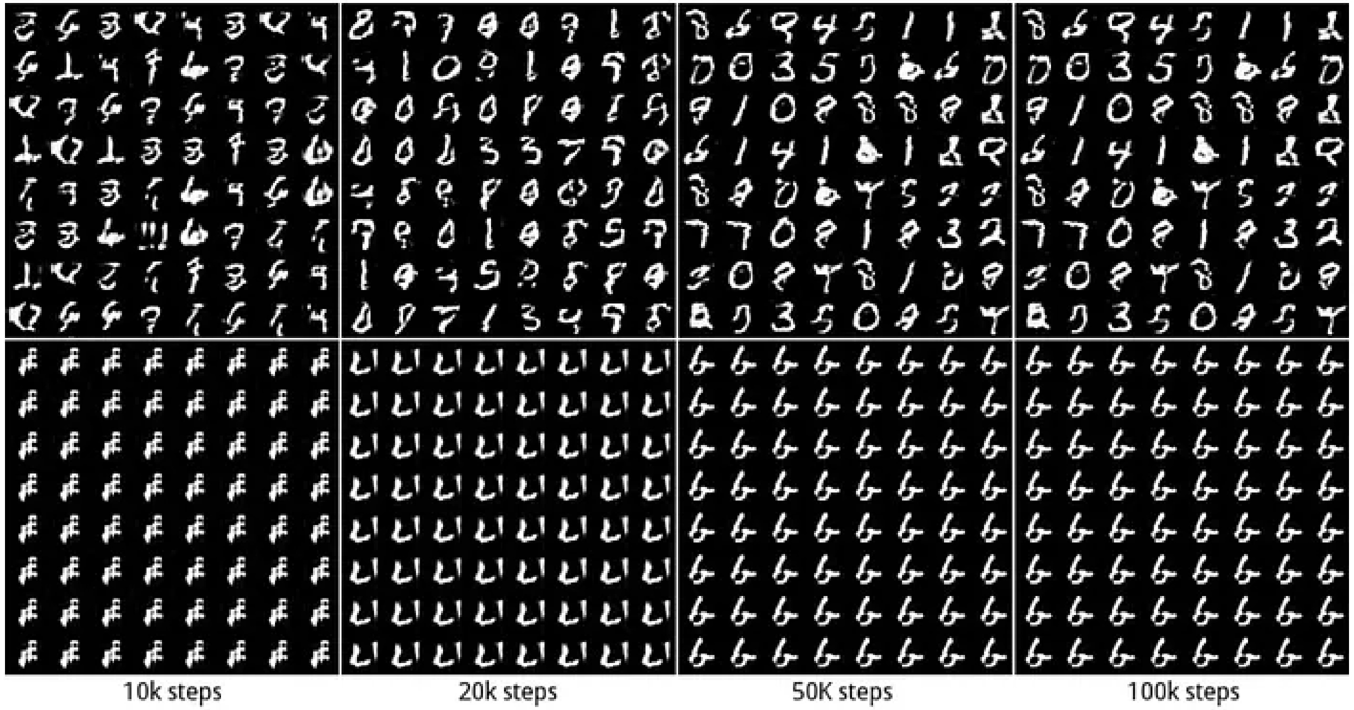
Mode

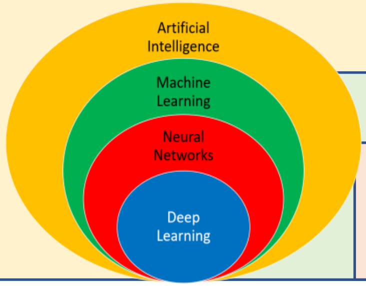
- Real-life data distributions are multimodal.

- For example, in MNIST, there are 10 major modes from digit ‘0’ to digit ‘9’. The samples below are generated by two different GANs.

- The top row produces all 10 modes while the second row creates a single mode only (the digit “6”).

- This problem is called mode collapse when only a few modes of data are generated.

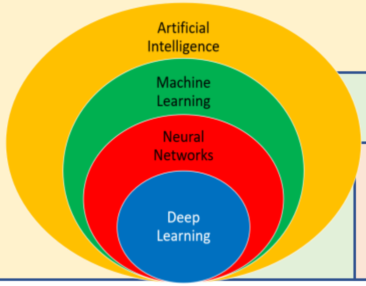




Nash equilibrium

- GAN is based on the zero-sum non-cooperative game.
- In short, if one wins the other loses.
- A zero-sum game is also called minimax. Your opponent wants to maximize its actions and your actions are to minimize them.
- In game theory, the GAN model converges when the discriminator and the generator reach a Nash equilibrium.
- This is the optimal point for the minimax equation below.

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



Nash equilibrium

- The Nash equilibrium refers to a scenario in which there exists no motivation for players to stray from their initial strategy alone.

- Consider two player A and B which control the value of x and y, respectively.

- Player A wants to maximize the value xy while B wants to minimize it.

$$\min_{B} \max_{A} V(D, G) = xy$$

- The **Nash equilibrium is x=y=0**. This is the state where the change of mind of a single player will not improve the result. Let's see whether we can find the Nash equilibrium easily using gradient descent.

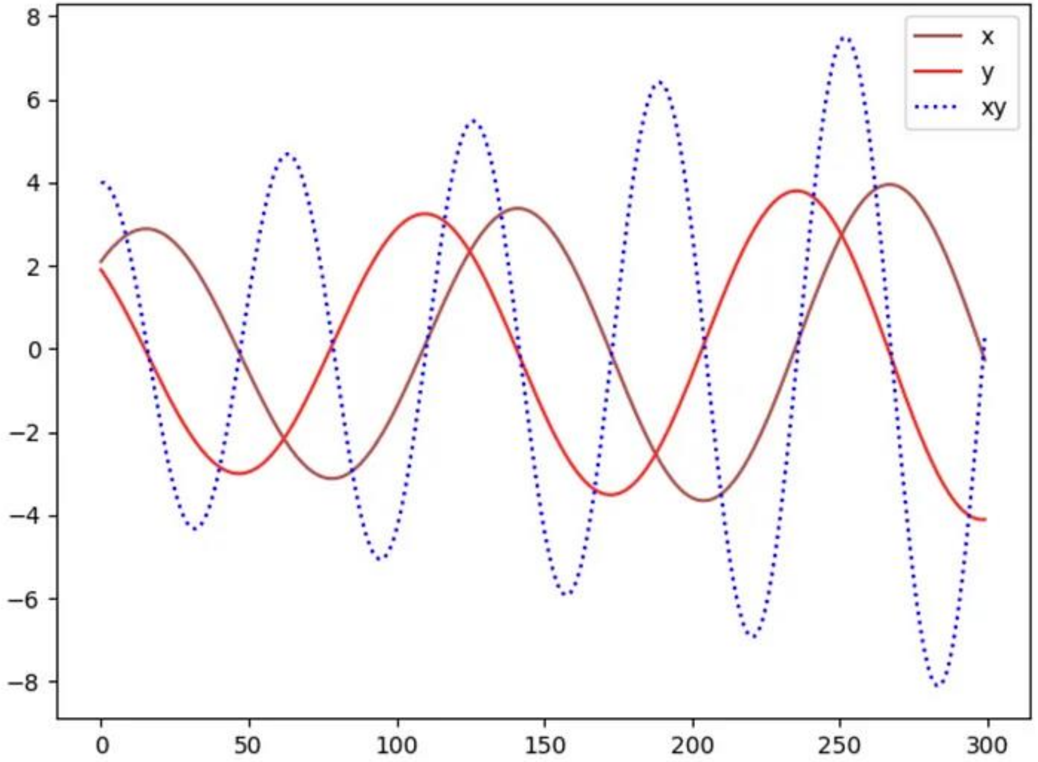
Nash equilibrium

- We update the parameters x and y based on the gradient of the value function V (α is the learning rate).

$$\Delta x = \alpha \frac{\partial(xy)}{\partial(x)}$$

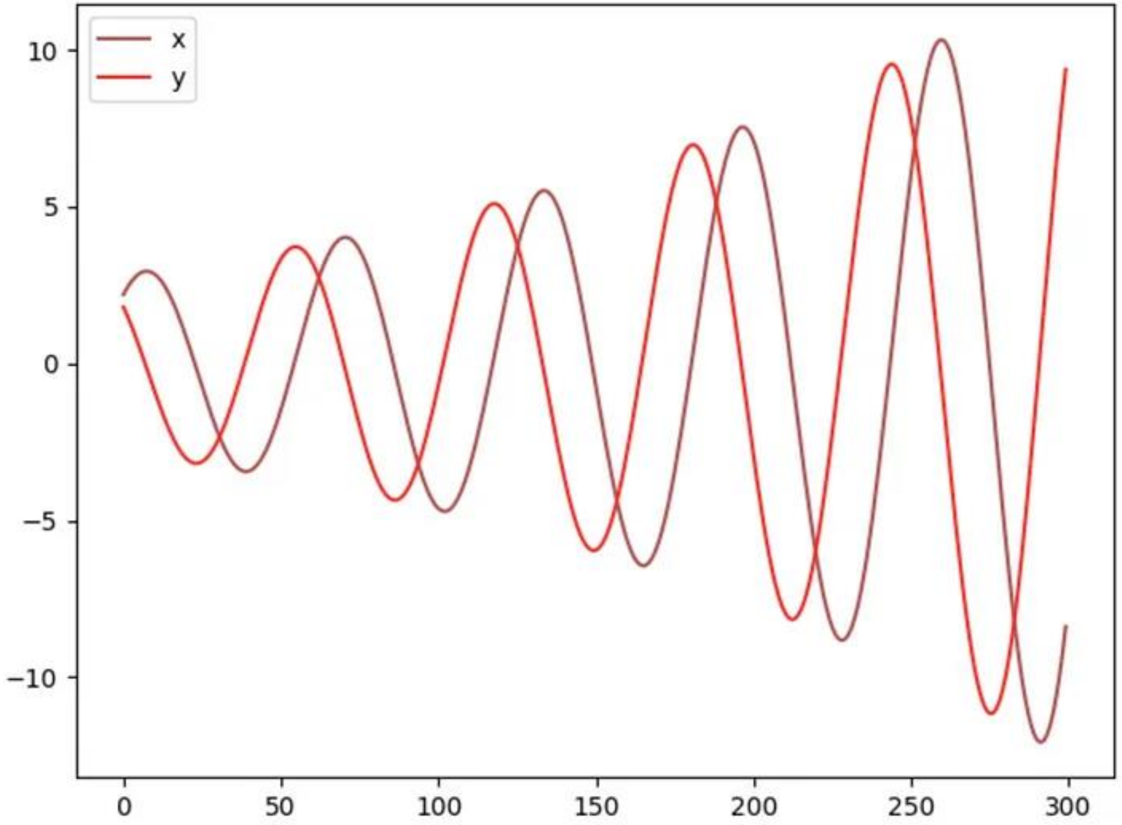
$$\Delta y = - \alpha \frac{\partial(xy)}{\partial(y)}$$

- When we plot x, y, and xy against the training iterations, we realize our solution does not converge.



Nash equilibrium

- If we increase the learning rate or train the model longer, we can see the parameters x, y is unstable with big swings.

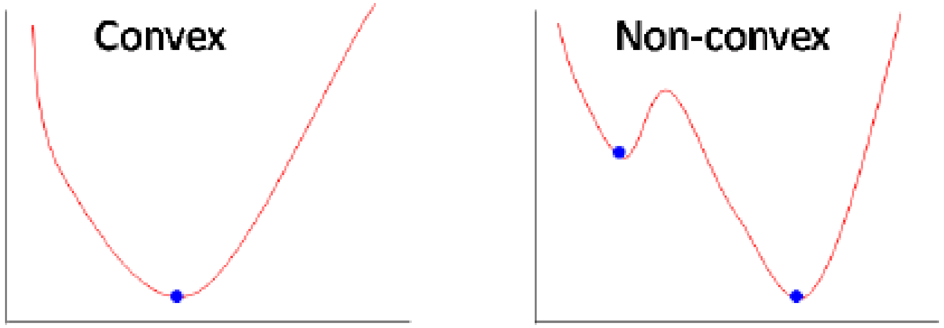


Nash equilibrium

- Our example is an excellent showcase that some cost functions will not converge with gradient descent, in particular for a non-convex game.

- We can also view this issue in an intuitive way: your opponent always countermeasures your actions which makes the models harder to converge.

- Cost functions may not converge using gradient descent in a minimax game.

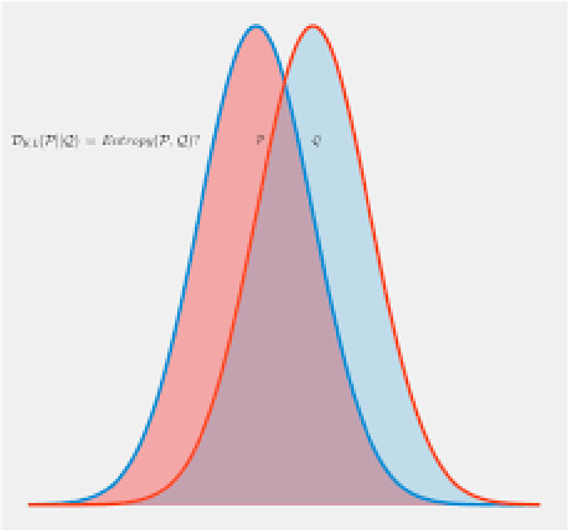


Generative model with KL-Divergence

- Our example is an excellent showcase that some cost functions will not converge with gradient descent, in particular for a non-convex game.

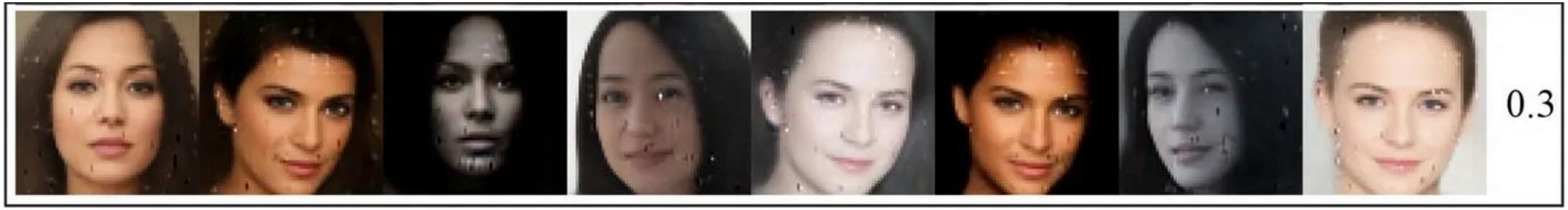
- We can also view this issue in an intuitive way: your opponent always countermeasures your actions which makes the models harder to converge.

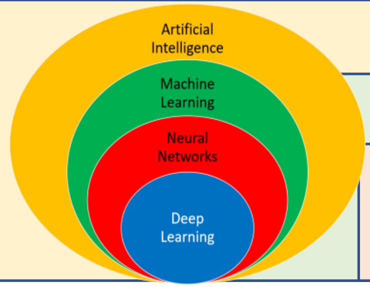
- Cost functions may not converge using gradient descent in a minimax game.



Why mode collapse in GAN?

- Mode collapse is one of the hardest problems to solve in GAN.
- A complete collapse is not common but a partial collapse happens often.
- The images below with the same underlined color look similar and the mode starts collapsing.





Why mode collapse in GAN?

- Let's see how it may occur. The objective of the GAN generator is to create images that can fool the discriminator D the most.

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)$$

- But let's consider one extreme case where G is trained extensively without updates to D. The generated images will converge to find the optimal image x* that fool D the most, the most realistic image from the discriminator perspective. In this extreme, x* will be independent of z.

$$x^* = argmax_x D(x)$$

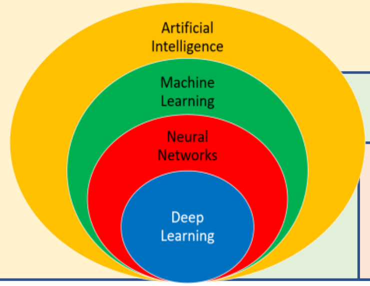
- This is bad news. The mode collapses to a **single point**. The gradient associated with z approaches zero.

$$\frac{\partial J}{\partial z} \approx 0$$

Why mode collapse in GAN?

- But mode collapse is not all bad news. In style transfer using GAN, we are happy to convert one image to just a good one, rather than finding all variants. Indeed, the specialization in the partial mode collapse sometimes creates higher quality images.

- But mode collapse remains one of the most important issues to be solved for GAN.



Hyperparameters & training

- No cost functions will work without good hyperparameters and tune them takes time and a lot of patience.
- New cost functions may introduce hyperparameter(s) that has sensitive performance.
- Hyperparameter tuning needs patience. No cost functions will work without spending time on the hyperparameter tuning.

Balance between the discriminator and generator

- The non-convergence and mode collapse are often explained as an imbalance between the discriminator and the generator.

- The obvious solution is to balance their training to avoid overfitting.

- However, very few progress has made but not because of the lack of trying.

- Some researchers believe that this is not a feasible or desirable goal since a good discriminator gives good feedback.

- Some of the attention is therefore shifted for cost functions with non-vanishing gradients instead.

Cost v.s. Image quality

- In a discriminative model, the loss measures the accuracy of the prediction and we use it to monitor the progress of the training.

- However, the loss in GAN measures how well we are doing compared with our opponent.

- Often, the generator cost increases but the image quality is actually improving.

- We fall back to examine the generated images manually to verify the progress.

- This makes model comparison harder which leads to difficulties in picking the best model in a single run. It also complicates the tuning process.

Conclusion

- GANs' applications have increased rapidly, in particular for images.
- GANs can be very interesting for companies.
- For example, GANs can generate realistic images of new medical images and image-to-image translation can help designers draw and be more creative.
- Moreover, GANs can be used for data augmentation when we only have one hundred images and we wish to have more.

Conclusion

- GANs have also been developed for binary outputs (sick or not) or discrete outputs (rounded blood pressure, rounded weight…).

- Benefits from this new research on tabular data are numerous, in particular for privacy purposes.

- For example, instead of sending confidential data from Excel sheets, hospitals can send fake realistic data (that keeps the correlation between the features) to their partners.