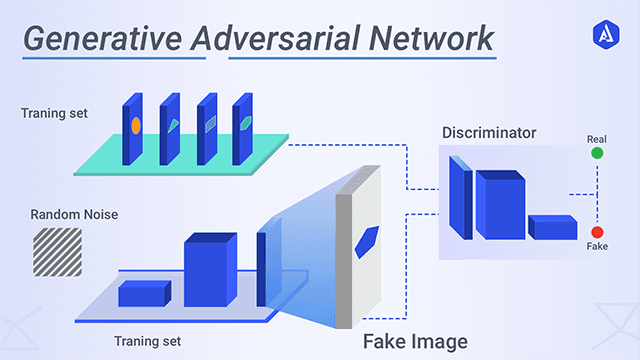
**Generative Adversarial Networks**

(Mạng đối nghịch tạo sinh - GAN)



**Giới thiệu**

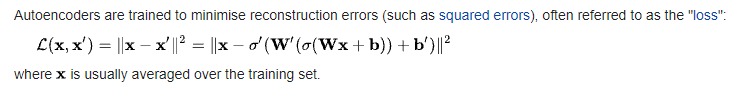
**Mạng đối nghịch tạo sinh** được biết đến là một mô hình máy học giúp cho máy tính có khả năng sáng tạo và suy nghĩ bằng việc tạo ra các dữ liệu trừu tượng giống như dữ liệu thật đến ngay cả con người không phân biệt được. Mạng được Ian Goodfellow giới thiệu vào năm 2014 thuộc nhóm **Generative Model** trong lĩnh vực máy học, điểm đặc biệt mạng sử dụng 2 mô hình học máy tương tác với nhau để hoàn thành việc huấn luyện mô hình chung.

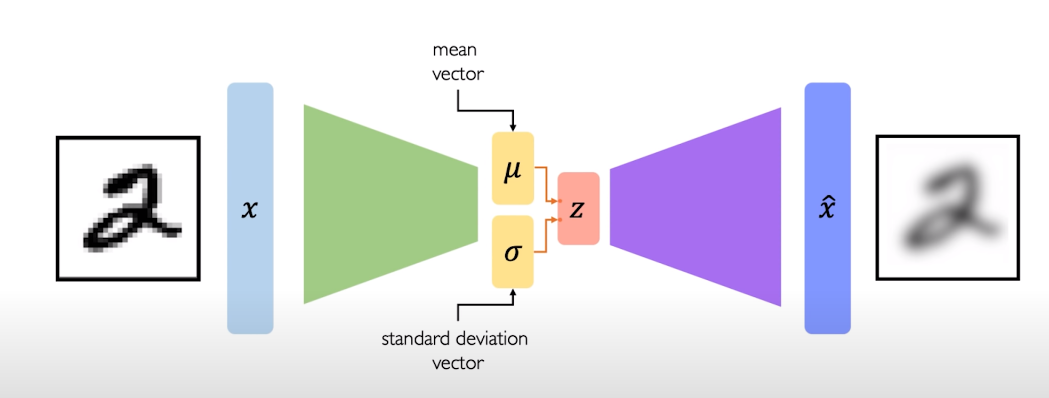
# Generative Model.

Là những mô hình máy học có khả năng tạo ra dữ liệu bằng việc học trên tập dữ liệu đã được cung cấp. Generative model bao gồm các mô hình máy học có giám sát (supervised) và không giám sát (unsupervised), trong đó có 2 mô hình nổi bật nhất là **Variational**-**Autoencoders** và **Generative adversarial network.**

## Autoencoders.

Encoder: Sử dụng các Deep layer để chuyển hóa dữ liệu đầu vào thành vector (Low Dimension Vector) ở chiều thấp hơn dữ liệu ban đầu hay nói cách khác là nén dữ liệu ban đầu thành vector đặc trưng.

Decoder: Sử dụng các Deep layer để tái cấu trúc lại dữ liệu ban đầu với đầu vào nhận vào là vector đặc trưng đã được Encoder.

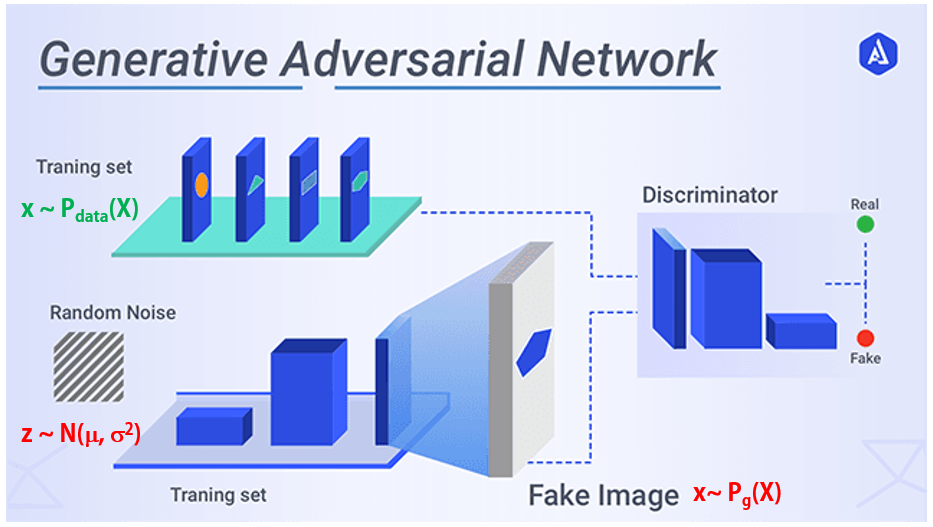
Variational autoencoders.

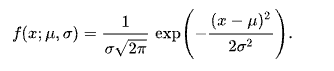
# Mạng đối nghịch tạo sinh.

## Kiến trúc mạng.

Mạng bao gồm 2 mô hình Neural networks tương tác với nhau là Generator và Discriminator.

*“GANs are based on a minimax game rather than an optimization problem, and have a value function that one agent seeks to maximize and the other seeks to minimize”*

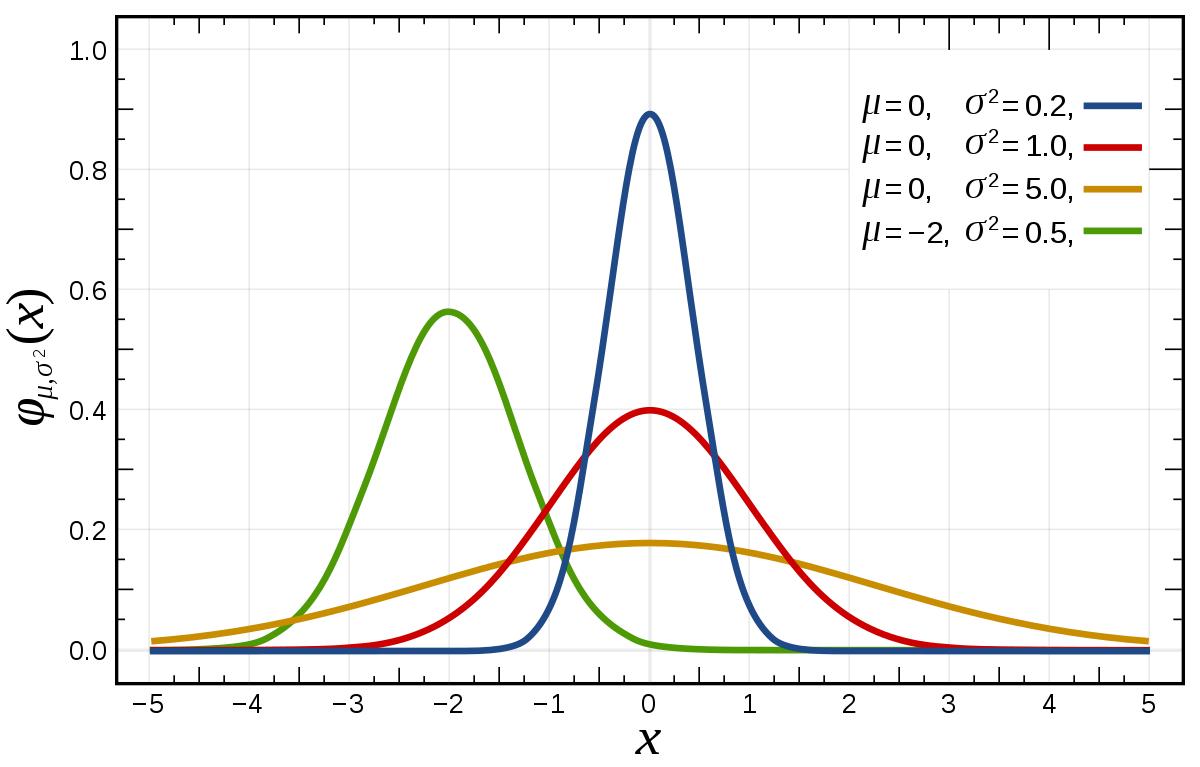


**Phân phối chuẩn (Phân phối tự nhiên hay Phân phối Gaussian)** là phân phối xác suất có hàm mật độ:

Trong đó μ là giá trị kỳ vọng (Trung bình), σ là độ lệch chuẩn (căn bậc 2 của phương sai).

Note: *Phân phối chuẩn có một tính chất quan trọng đó là “Định lý giới hạn trung tâm”, khi số lượng mẫu quan sát ta có được tiến tới vô cùng thì phân phối biểu diễn cho tập dữ liệu đó tiến về phân phối chuẩn, điều này chứng minh rằng phân phối chuẩn là phân phối biểu diễn tổng quá cho các tập dữ liệu tự nhiên.*

*N(số lượng mẫu) → ∞ => Pdata → N(μ,σ)*



Mạng sẽ học cách tối ưu hóa sao cho Pdata = Pg thông qua trò chơi minimax giữa Generator và Discriminator, khi mà cả 2 cùng đạt trạng thái **Nash-equilibrium**.

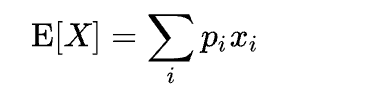
*“D and G play the following two-player minimax game with value function V(G, D)”*

**Loss Function:**

Trong đó:

* D(x): Giá trị dự đoán của Discriminator đối với dữ liệu thật (x ~ Pdata(x)).
* D(G(z)): Giá trị dự đoán của Discriminator đối dữ liệu được tạo bởi Generator (z ~ Pz(x)).
* **E**x~p[x]: Giá trị trung bình của X.

Note: *Nếu X là một biến ngẫu nhiên rời rạc thì* ***E****[x] được tính:*

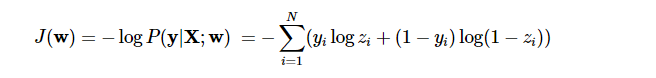
**

*Trong đó pi­ (đôi khi ký hiệu là f(x) là hàm mật độ xác suất) là xác suất xuất hiện của giá trị Xi*

*Nếu X là một biến ngẫu nhiên liên tục thì được tính:*

**Giải thích hàm loss:**

Logistic Regression Loss (Label: 1 – Real data, 0 – Fake data):



Trong Logistic Regression chúng ta cố gắng tối ưu hóa hàm Loss bằng việc đi tìm Global Minimum bằng thuật toán Gradient decent.

Tối ưu hóa hàm Loss bằng việc đi tìm Global Maximum và sử dụng Gradient ascent, vì dữ liệu đã được gán nhãn trước (1 cho dữ liệu thật và 0 cho dữ liệu của Generator), nên ta có thể tối giản hàm Loss như sau:

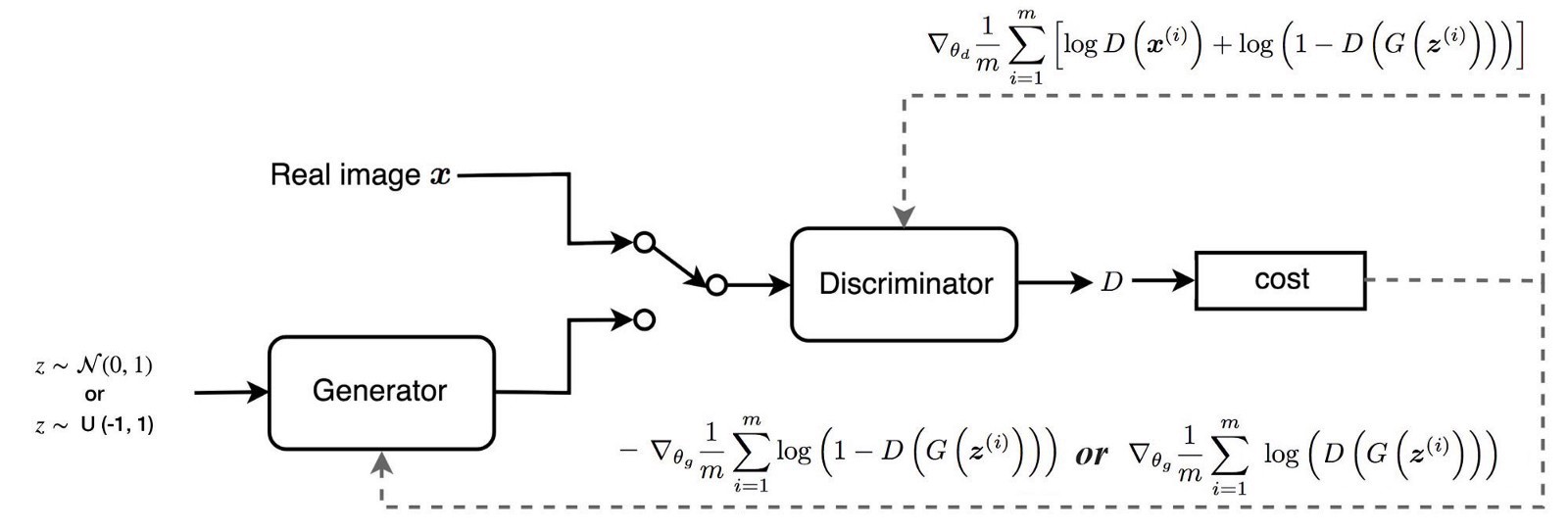
J(**w**) = logP(y| **X**;**w**) = )

*Việc thế Label yi = 1 và yi = 0 thì hàm loss ta được tối giản như trên*

“We train D to maximize the probability of assigning the correct label to both training examples and samples from G. We simultaneously train G to minimize log(1 − D(G(z)))”

*Note: Minimize Log(1 – D(G(z))) có thể thay thế bằng việc Maximize Log(D(G(z))) (Việc maximize Log(y) đồng nghĩa với việc muốn mô hình học cách tăng xác suất giá trị y hay giá trị dự đoán xác suất của label y)*

Cả 2 mạng sẽ học cho đến khi mô hình đạt trạng thái cân bằng (**Nash-equilibrium)**, Discriminator có học thêm nữa cũng không thể cải thiện việc phân biệt giữa dữ liệu thật và dữ liệu giả (D(x) = 0.5), Generator có học thêm nữa cũng không thể đánh lừa tiếp được Discriminator.

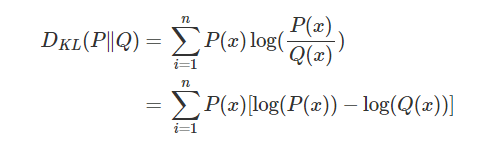


## Hội tụ.

*Chứng minh sự hội tụ của mô hình hay cách mô hình học để dữ liệu được tạo ra trong giống với dữ liệu thật, không cần thiết phải tìm hiểu sâu phần này*

Cross entropy thể hiện sự tương quan giữa 2 phân phối dữ liệu (Pdata(x), Pg(x)) nên được sử dụng làm hàm Loss trong GAN, việc hội tụ Cross Entropy cũng đồng nghĩa với việc hội tụ 2 phân kỳ *Kullback-Leibler Divergence* và *Jensen-Shannon Divergence*, đây là 2 phân kỳ thể hiện sự tương đồng giữa 2 phân phối dữ liệu, phân kỳ càng nhỏ 2 phân phối càng giống nhau và ngược lại.

Kullback-Leibler Divergence:



Ta có thể suy ra:



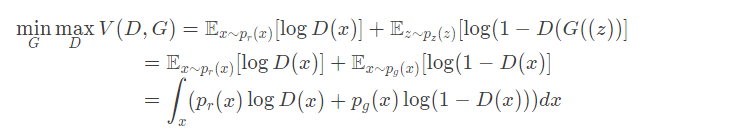
Phân kỳ Kullback-leibler là một hàm bất đối xứng (DKL (P||Q) **≠** DKL (Q||P))

Jensen-Shannon Divergence: Khắc phục nhược điểm tính bất đối xứng của phân kỳ Kullback-leibler, bằng việc tính phân kỳ Kullback-leibler so với phân phối trung bình giữa cả 2.

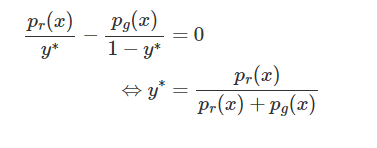
*DJS (P||Q) = DKL(P||) – DKL(Q||)*

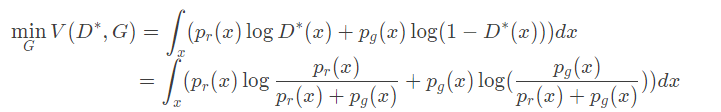
Ta có thể suy ra:

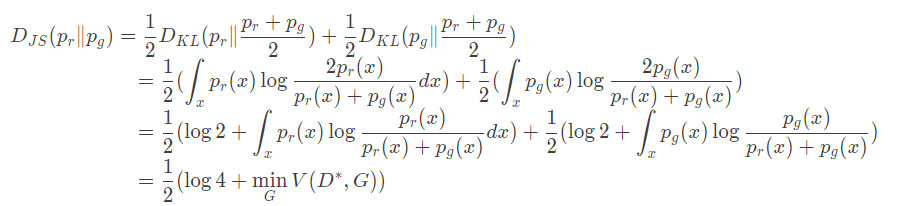
*DJS (P||Q) =* ***H****() –(* ***H****(P) –* ***H****(Q))*

Xét hàm Loss của GAN, coi G là hằng số suy ra hàm Loss sẽ bằng:

Ta có thể tìm cực đại của hàm Loss bằng việc tìm nghiệm của đạo hàm bậc nhất bằng 0 khi đó ta được:



Trong đó y\* = D(x), thay vào hàm Loss gốc ta được:

 Từ phân kỳ *Jensen-Shannon* và công thức trên ta có thể suy ra:

* 2DJS (Pr||Pg) –log(4)
* Hàm số đạt cực tiểu (Global minimum) tại giá trị -2log(2) khi mà Pr = Pg, các phân kỳ giá trị không thể nhỏ hơn 0.

# Biến thể GAN.

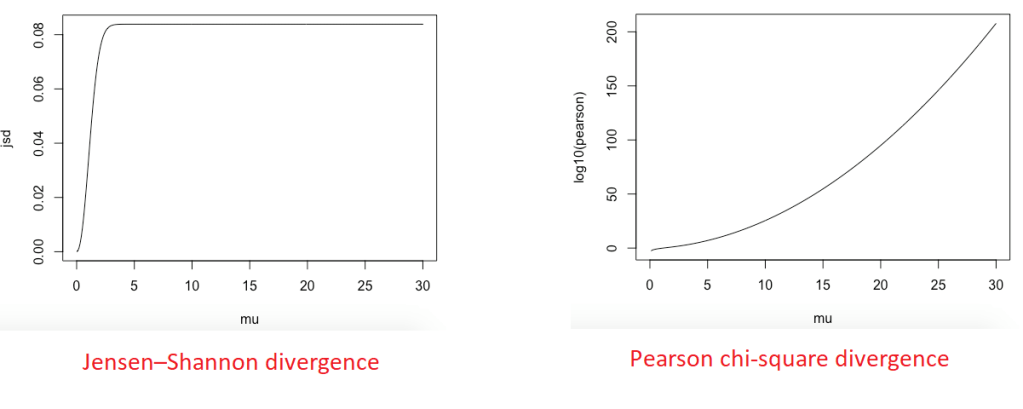
## https://miro.medium.com/max/1400/0*d6uK5r08HQEhWbLW.pngDeep Convolution GAN (DCGAN).

## https://machinelearningmastery.com/wp-content/uploads/2019/05/Example-of-a-Conditional-Generator-and-a-Conditional-Discriminator-in-a-Conditional-Generative-Adversarial-Network-1024x887.pngCondition GAN.

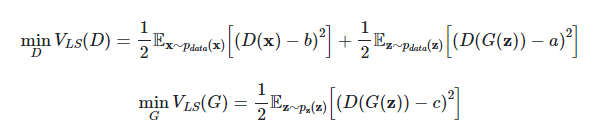
1 2 3, = 1 =1

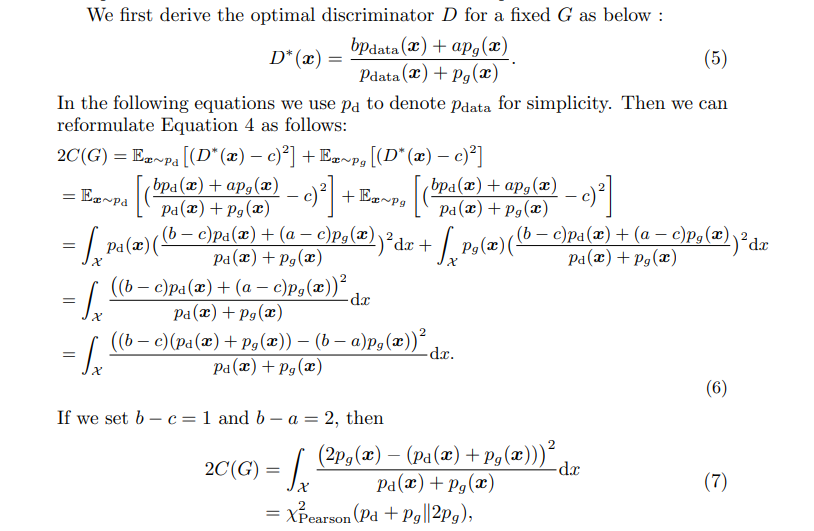
2=2, 3=3.

## Least Square GAN.

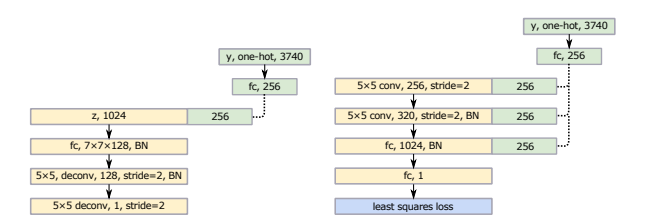


Loss Least Square GAN:

Trong đó a, b, c là các target Label mong muốn.



Trong điều kiện trên thì a = -1, b = 1, c = 0.



## Wasserstein GAN.

