ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN CUỐI KỲ

MATH FOR COMPUTER SCIENCE

Đề tài: GENERATIVE ADVERSARIAL NETWORKS

LỚP: CS115.M11 – Khoa học Máy tính

GVHD: TS. Lương Ngọc Hoàng

THÀNH VIÊN:

1. Nguyễn Hoài Nam - 20520075
2. Bùi Nguyễn Anh Trung - 20520332
3. Đoàn Phương Khanh - 20521443
4. Lê Nguyễn Minh Huy - 20521394
5. Vũ Thị Phương Linh - 20521541

TP. Hồ Chí Minh - 12/2021

MỤC LỤC

1. ĐẶT VẤN ĐỀ

Hiện nay, Deep Learning đã tạo ra một bước tiến vượt bậc, cung cấp khả năng nhận diện hình ảnh và âm thanh, hiểu biết ngôn ngữ tự nhiên cùng nhiều tác vụ khác có thể thay thế con người.

Tuy nhiên, để huấn luyện được một mô hình tốt cùng với độ chính xác cao thì nguồn dữ liệu chính là phần đóng vai trò quan trọng và ảnh hưởng lớn nhất. Chúng ta không thể sử dụng làm dữ liệu huấn luyện và đánh giá nếu không có sự quan sát, chọn lọc và gán nhãn tỉ mỉ của con người. Điều này rất tốn thời gian và công sức.

Vì nhu cầu đó, bài toán Generator ra đời với mục đích sinh dữ liệu giả có độ tin cậy tương đối để phục vụ cho các mô hình huấn luyện. Generative Adversarial Networks (GAN)**[1]** là một phương pháp sinh dữ liệu ra đời vào năm 2014, nhưng cho đến hiện nay vẫn chứng minh được sự hiệu quả bên cạnh các phương pháp phức tạp và hiện đại hơn. Nó chính là một trong những kỹ thuật thể hiện sự sáng tạo trong trí tuệ nhân tạo.

2. TỔNG QUAN ĐỀ TÀI

Generative Adversarial Networks lấy ý tưởng từ Minimax Game **[8],** được hiểu như là trò chơi cạnh tranh giữa hai người. Tất cả đều mong muốn cải thiện điểm số của mình để dành chiến thắng.

Trong bài báo cáo này, chúng tôi tập trung vào việc tìm hiểu, phân tích đầy đủ mặt toán học của thuật toán GAN và phương pháp đánh giá Inception Score **[3]** dưới lý thuyết tương trợ thông tin (Mutual information)**[9].** Đây là điều chưa được thực hiện đồng thời ở các bài báo liên quan.

Bên cạnh đó, chúng tôi cũng thực hiện đánh giá so sánh giữa hai phương pháp GAN và Deep Convolutional GAN (DCGAN)**[2]** trên bộ dữ liệu Fashion MNIST**[4]** và ANIME**[10]** để đưa kết luận về độ hiệu quả cùng với những vấn đề chưa được khắc phục của GAN.

3. MINIMAX GAME

Trong lý thuyết trò chơi, giả sử người chơi i chọn chiến lược s_i và những người còn lại là chiến lược s_{-i}. Gọi $v_i(s)$ là hàm lợi ích của người chơi i trong chiến lược s, Minimax của trò chơi được định nghĩa

$$\underline{v_i} = v_i(s_i, s_{-i})$$

Tương tự ta cũng có Maximin được định nghĩa bởi:

$$\underline{v_i} = v_i(s_i, s_{-i})$$

Định lý Minimax của Von Neumman

Tiếp theo, ta sẽ chứng minh Minimax theo Von Neumman**[11]** được định nghĩa như sau:

Cho A là ma trận $m \times n$ đại diện cho ma trận chi phí của hai người chơi. Khi đó trò chơi sẽ luôn tồn tại một giá trị và một cặp chiến lược hỗn hợp tối ưu cho hai người chơi. Với x, y là các vector xác xuất tương ứng cho mỗi người chơi khi ra quyết định, ta có các cặp chiến lược hỗn hợp $(x, y) = \big((x_1, x_2, \dots, x_m), (y_1, y_2, \dots, y_n)\big)$

Với mỗi cặp chiến lược (x, y) ta định nghĩa:

$$V(x, y) \coloneqq \sum_{i=1}^{m} \sum_{j=1}^{n} x_i, a_{ij}, y_j,$$

Trong đó a_{ij} là chi phí tại vị trí (i, j) trong A.

Ta gọi một cặp chiến lược hỗn hợp (x^*, y^*) là một cặp cân bằng (equilibrium point) cho một trò chơi Minimax game hai người thoả mãn:

$V(x, y^*) \leq V(x^*, y^*)$ với mọi $x \in X_m$, và $V(x^*, y^*) \leq V(x^*, y)$ với mọi $y \in Y_n$

Điều này tương đương với:

$$V(x, y^*) = V(x^*, y^*) = V(x^*, y)$$

Khi cân bằng xảy ra, xác suất thắng của hai người chơi là như nhau. Do đó ta cần chứng minh sẽ luôn tồn tại cặp cân bằng (x^*, y^*).

4. GENERATIVE ADVERSARIAL NETWORKS

4.1. Ý tưởng hoạt động của GAN

Ý tưởng cơ bản của GAN là thiết lập một trò chơi hai người. Một trong số đó là Generator (module G), Generator tạo ra các mẫu (samples) từ không gian ngầm (latent space) có cùng phân phối với dữ liệu huấn luyện. Người chơi còn lại là Discriminator (module D), người này đánh giá các samples và xác định xem các samples là thật hay giả.

4.2. Latent Space

Trong thống kê, các biến tiềm ẩn (latent variables) là các biến không được quan sát (thông qua phép đo) trực tiếp nhưng được suy ra (thông qua mô hình toán học) từ các biến khác đã được quan sát.

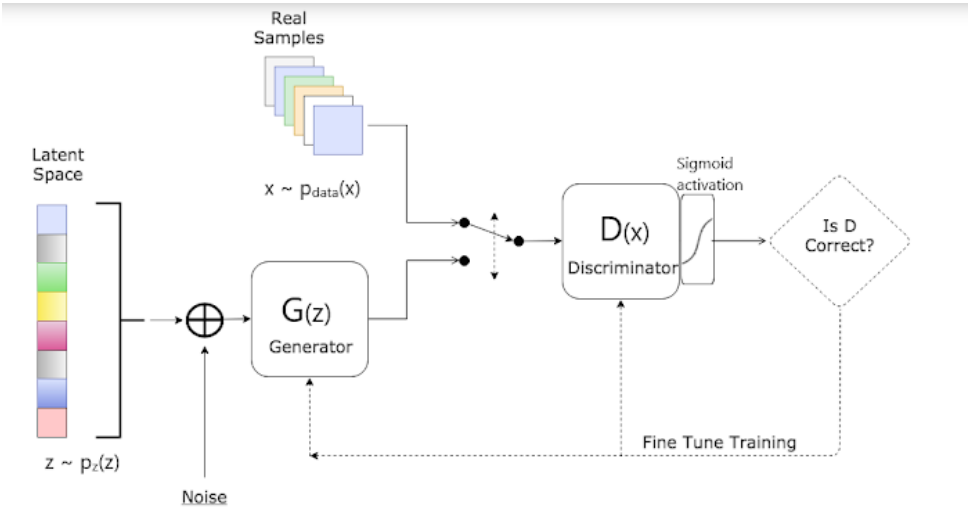
Khi đó, latent space **[12]** là không gian của latent variables, đề cập đến một không gian đa chiều trừu tượng chứa các đặc trưng mà chúng ta không thể giải thích trực tiếp.

Với Deep Learning, ý tưởng của Latent Space là quan trọng vì nó giúp học các đặc trưng của dữ liệu và đơn giản hoá các biểu diễn của dữ liệu cho mục đích tìm kiếm các mẫu.

4.3. Generator and Discriminator

Một số ký hiệu sử dụng:

- X, Y, Z: lần lượt là không gian miền dữ liệu thật, dữ liệu giả và dữ liệu nhiễu
- G: Generator
- D: Discriminator
- θ_g: Tham số Generator
- θ_d: Tham số của Discriminator
- $p_{data}(x)$: phân phối của dữ liệu thật
- $p_z(z)$: phân phối của dữ liệu nhiễu
- $p_g(x)$: phân phối của dữ liệu giả tạo ra bởi G



Hình 1: Kiến trúc mô hình GAN

4.3.1. Mặt toán học

- $G: Z \to Y$ là một hàm khả vi đi từ Latent Space Z vào Y, được đại diện bởi một multilayer perceptron với các tham số θ_g
- $D: X \cup Y \to [0,1]$ là một hàm xác xuất đi từ không gian Y hoặc X, đại diện bởi một multilayer perception với các tham số θ_d

4.3.2. Loss function

Loss function trong GAN

$$V(G,D) = E_{z\sim p_{data}}[logD(x)] + E_{z\sim p_z(z)}[log(1 - D(G(z)))]$$

Rõ ràng mục tiêu của Discriminator là maximize giá trị $D(x)$ đối với những điểm dữ liệu $x \sim p_{data}(x)$ và minimize $D\big(G(z)\big)$ đối với những điểu dữ liệu $z \sim p_z(z)$ hay maximize $1 - D\big(G(z)\big)$

Ta thấy rằng, $maxU \Leftrightarrow maxlogU, \forall 0 < U \leq 1.$

Khi đó, mục tiêu của D là maximize tổng:

$$E_{z\sim p_{data}}[logD(x)] + E_{z\sim p_z(z)}[log(1 - D(G(z)))] = V(G, D)$$

Ký hiệu D tối ưu là $D_G^* = argmax_D V(G, D)$, với G cho trước.

Giả sữ đã có $D = D_G^*$, do mục tiêu của G và D là ngược nhau, ta tìm G để minimize $V(G, D)$, lúc này G đã tối ưu được ký hiệu là

$$G^* = argmin_G V(G, D_G^*)$$

Ta đi chứng minh bài toán tối ưu có nghiệm duy nhất G^* thoả điều kiện $p_g = p_{data}$.

Theo định lý Radon-Nikodym **[13],** ta có:

$$V(G, D) = \int_z p_{data} log\big(D(x)\big)dx + \int_z p_z(z) log\left(1 - D\big(G(z)\big)\right)dz$$

$$= \int_z p_{data} log\big(D(x)\big) + p_g(x) log\big(1 - D(x)\big)dx$$

Ta có $\big(p_{data}, p_g\big) \neq (0,0)$, để tìm maximun $V(G, D)$, ta thực hiện như sau:

$$\frac{dV(G, D)}{dx} = 0 \Leftrightarrow \frac{p_{data}}{D(x)} - \frac{p_g(x)}{1 - D(x)} = 0 \Leftrightarrow D(x) = \frac{p_{data}}{p_{data} + p_g(x)}$$

$$\frac{d^2V(G, D)}{dx^2} = -\frac{p_{data}}{D^2(x)} - \frac{p_g(x)}{\big(1 - D(x)\big)^2}$$

Thay biểu thức của $D(x)$ vào ta được:

$$\frac{d^2V(G,D)}{dx^2} < 0 \Leftrightarrow -\frac{\big(p_{data} + p_g(x)\big)^2}{p_{data}} - \frac{p_g(x)}{\left(1 - \frac{p_{data}}{p_{data} + p_g(x)}\right)^2} < 0 \text{ (luôn đúng)}$$

Vậy $D(x) = \frac{p_{data}}{p_{data}+p_g(x)}$ là maximum của tích phân trên, hay $D_G^* = \frac{p_{data}}{p_{data}+p_g(x)}$

Khi đó, nếu gọi $C(G)$ là mục tiêu huấn luyện dữ liệu giả (virtual training criterion)

$$C(G) = V(G, D)$$

$$C(G) = E_{x\sim p_{data}}[logD_G^*(x)] + E_{z\sim p_z}\left[log\left(1 - D_G^*\big(G(z)\big)\right)\right]$$

$$C(G) = E_{x\sim p_{data}}[logD_G^*(x)] + E_{x\sim p_g}\left[log\big(1 - D_G^*(x)\big)\right]$$

$$C(G) = E_{x\sim p_{data}}\left[log\frac{p_{data}(x)}{p_{data}(x) + p_g(x)}\right] + E_{x\sim p_g}\left[log\frac{p_g(x)}{p_{data}(x) + p_g(x)}\right]$$

4.4. Định lý quan trọng nhất của thuật toán GAN

Global minimum của $C(G)$ đạt được khi và chỉ khi $p_{data}(x) = p_g(x)$.

Khi đó $minC(G) = -log(4)$

Trước khi chứng minh, ta nhắc lại định nghĩa độ đo Kullback-Leinler divergence (KL) **[14]** và Jensen-Shannon divergence (JSD) **[15]**

Với p, q là các phân phối xác suất trên trên cùng một không gian xác xuất X. Độ đo Kullback-Leibler divergence từ q vào p được định nghĩa:

$$KL(|q) = \sum_{x\in X} \quad p(x)log\left(\frac{p(x)}{q(x)}\right)$$

Độ đo Jensen-Shannon divergence được xác định là:

$$JSD(|q) = \frac{1}{2}KL(|m) \ + \frac{1}{2}KL(|m)$$

Đây là độ đo không âm nên $JSD(q) \geq 0$.

Chứng minh:

Ta biến đổi biểu thức $C(G)$ như sau:

$$C(G) = E_{x\sim p_{data}}\left[log(2) \ - \ log(2) \ + \ log\frac{p_{data}(x)}{p_{data}(x) + p_g(x)}\right] + E_{x\sim p_g}\left[log(2) \ - \ log(2) \ + \ log\frac{p_g(x)}{p_{data}(x) + p_g(x)}\right]$$

$$C(G) = -2log(2) + E_{x\sim p_{data}}\left[log(2) + log\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}\right] + E_{x\sim p_g}\left[log(2) + log\frac{p_g(x)}{p_{data}(x)+p_g(x)}\right]$$

$$C(G) = -2log(2) + E_{x\sim p_{data}}\left[log\frac{p_{data}(x)}{(p_{data}(x)+p_g(x))/2}\right] + E_{x\sim p_g}\left[log\frac{p_g(x)}{(p_{data}(x)+p_g(x))/2}\right]$$

$$C(G) = -log(4) + 2JSD(|p_g) \geq -log(4)$$

Dấu " = " xảy ra khi và chỉ khi $p_{data}(x) = p_g(x)$.

Khi đó $C^* = min_G C(G) = -log(4)$

4.5. Thuật toán của GAN

for *number of training iterations* **do**

for *k steps* **do**

- Sử dụng *m* mẫu dữ liệu $z^{(1)}, \ldots\ldots, z^{(m)}$ được tạo ra bởi phân phối $p_g(z)$
- Sử dụng *m* mẫu dữ liệu $x^{(1)}, \ldots\ldots, x^{(m)}$ được tạo ra bởi phân phối $p_{data}(x)$
- Cập nhập tham số discriminator tăng theo stochastic gradient

$$\nabla_{\theta_d}\frac{1}{m}\sum_{i=1}^{m}\left[logD(x^{(i)}) + log\left(1 - D\left(G(z^{(i)})\right)\right)\right]$$

end for

- Sử dụng *m* mẫu dữ liệu $z^{(1)}, \ldots\ldots, z^{(m)}$ được tạo ra bởi phân phối $p_g(z)$
- Cập nhập tham số generator giảm theo stochastic gradient

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

end for

4.6. Mệnh đề về sự hội tụ

Nếu G và D đủ khả năng, và ở mỗi bước của thuật toán trên, Discriminator được tối ưu với G cho trước, và p_g được cập nhập để cải tiến

$$E_{x\sim p_{data}}[logD_G^*(x)] + E_{z\sim p_z}\left[log\left(1 - D_G^*(G(z))\right)\right]$$

Khi đó $p_g \rightarrow p_{data}$

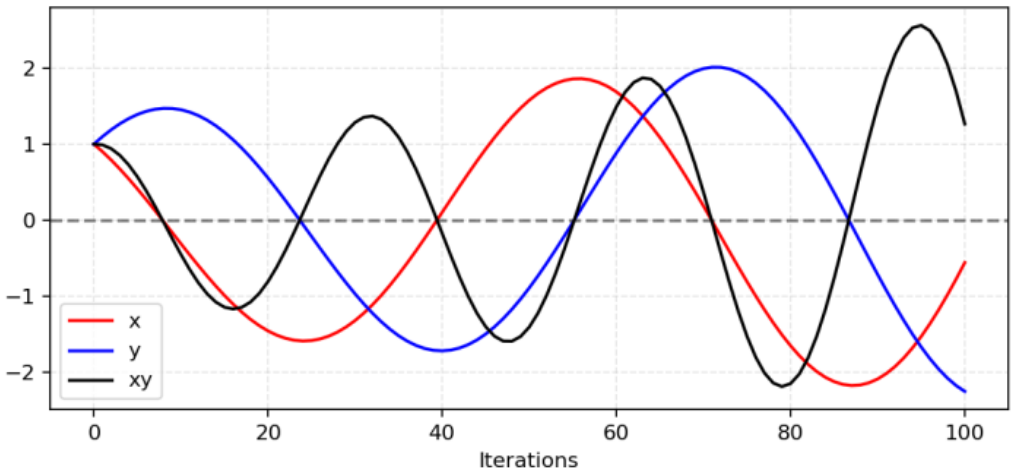
4.7. Một số vấn đề của GAN

4.7.1. Failure to Converge

Một trong những vấn đề đầu tiên của GAN đó chính là khó khăn trong việc hội tụ.

Công việc chính của GAN chính là đi tối ưu cả hai module G và D, đây là bài toán tối ưu hai biến nên rất khó đạt được hội tụ toàn cục.

Giả sử xét xy và xy thì điểm cân bằng sẽ là $(0,0)$ như hình vẽ bên dưới



Hình 2: Sự biến thiên của các hàm số x, y, xy

4.7.2. Vanishing Gradient

Nếu module Discriminator hoạt động quá tốt, thì điều nảy dẫn đến việc module Generator xảy ra hiện tượng vanishing gradient

Cụ thể tại những step đầu tiên thì Generator hoạt động chưa tốt, khi đó sẽ tạo ra sự khác biệt lớn giữa dữ liệu fake (giả) và real (thật). Điều này làm cho Discriminator hoạt động tốt và dẫn đến sự hội tụ nhanh chóng. Tuy nhiên Generator sẽ gặp thất bại trong quá trình huấn luyện, vì nó không thể học thêm được gì khi xảy ra hiện tượng vanishing gradient.

4.7.3. Mode collapse

Hiện tượng Mode collapse xảy khi Generator phát hiện điểm mù của Discriminator. Đây là điểm mà Discriminator không thể phân biệt được thật/giả, vì thế Generator sẽ luôn cố

học và tạo ra dữ liệu dựa vào điểm mù đó. Gây ra hiện tượng mode collapse, làm cho tập dữ liệu đầu ra giống hệt nhau.

5. TƯƠNG TRỢ THÔNG TIN

5.1. Khái niệm

Tương trợ thông tin (Mutual information) **[9]** là khái niệm cơ bản trong lý thuyết thông tin định lượng "lượng thông tin" dự kiến có trong một biến ngẫu nhiên.

Cụ thể hơn, nó định lượng "lượng thông tin" (theo các đơn vị như shannons (bit), nats hoặc hartleys) thu được về một biến ngẫu nhiên bằng cách quan sát biến ngẫu nhiên kia.

Cho (X,Y) là cặp biến ngẫu nhiên các giá trị thuộc không gian $X\ x\ Y$. Phân phối đồng thời của chúng là $P_{X,Y}$ và phân phối lề lần lượt là $P_x,\ P_y$. Khi đó, tương hỗ thông tin được định nghĩa là:

$$I(X;Y) = \ D_{KL}(P_{X,Y}||P_X \otimes P_Y) = \sum_{y\in Y} \ \sum_{x\in X} \ p_{X,Y}(x,y) log\ (\frac{p_{(X,Y)}(x,y)}{p_X(x)p_Y(y)})$$

5.2. Tính chất

Tính chất 1: Khi hai biến X,Y độc lập thì $p_{(X,Y)}(x,y) = p_X(x)p_Y(y)$.

Do đó $log\ log\left(\frac{p_{(X,Y)}(x,y)}{p_X(x)p_Y(y)}\right) = log\ log\ (1)\ = 0$.

Tức là việc biết X không hỗ trợ gì cho việc tìm hiểu Y và ngược lại. Do đó hỗ trợ thông tin của chúng bằng 0

Tính chất 2: Tính không âm :

Sử dụng bất đẳng thức có thể có thể chứng minh rằng $I(X;Y)$ không âm

Chứng minh: Theo bổ đề bất đẳng thức Jensen , với bất kỳ hàm lồi f(x) ta đều có:

$$E[f(x)] \geq f(E[x])$$

$$I(X;Y) = \ D_{KL}(P_{X,Y}||P_X \otimes P_Y)$$

Ta chỉ cần chứng minh $relative\ entropy$ với hai hàm phân phối $p(x)$ và $q(x)$ được định nghĩa là $D(p(x)||q(x))$ không âm thì mutual information cũng không âm

$$D(p(x)||q(x)) = \sum_{x} \ p(x)\ log\ log\ \frac{p(x)}{q(x)}$$

$$= -\sum_{x} \quad p(x)\ log\ log\ \frac{q(x)}{p(x)}$$

$$= -E[log\ log\ \tfrac{q(x)}{p(x)}] \geq -log\ log\ \left(E\left[\tfrac{q(x)}{p(x)}\right]\right)$$ (do hàm log là hàm lõm)

$$= -log\ log\left(\sum_{x} \quad p(x)\ log\ log\ \frac{q(x)}{p(x)}\right) = -log\ (\sum_{x} \quad q(x)) = 0$$

Tính chất 3: Tính đối xứng

$$I(X;Y) = \sum_{y\in Y} \quad \sum_{x\in X} \quad p_{X,Y}(x,y) log\ (\frac{p_{(X,Y)}(x,y)}{p_X(x)p_Y(y)})$$

$$I(Y;X) = \sum_{x\in X} \quad \sum_{y\in Y} \quad p_{X,Y}(x,y) log\ (\frac{p_{(X,Y)}(x,y)}{p_X(x)p_Y(y)})$$

Thế nên $I(X;Y) = I(Y;X)$

Tính chất 4: Biểu diễn theo phân rã Kullback-Leibler

$$I(X;Y) = I(Y;X) = \sum_{x\in X} \quad \sum_{y\in Y} \quad p_{X,Y}(x,y) log\ (\frac{p_{(X,Y)}(x,y)}{p_X(x)p_Y(y)})$$

$$= \sum_{y\in Y} \quad \sum_{x\in X} \quad p_{X|Y=y}(x) p_Y(y)\ log\ log\ \frac{p_{X|Y=y}(x) p_Y(y)}{p_X(x)p_Y(y)}$$

$$= \sum_{y\in Y} \quad p_Y(y) \sum_{x\in X} \quad p_{X|Y=y}(x)\ log\ log\ \frac{p_{X|Y=y}(x)}{p_X(x)}$$

$$= \sum_{y\in Y} \quad p_Y(y) D_{KL}(p_{X|Y=y} || p_X)$$

$$= E_Y[D_{KL}(p_{X|Y} || p_X)]$$

Tính chất 5: Biểu diễn qua Entropy điều kiện và Entropy đồng thời

$$I(X;Y) = I(Y;X) = \sum_{x\in X} \quad \sum_{y\in Y} \quad p_{X,Y}(x,y) log\ (\frac{p_{(X,Y)}(x,y)}{p_X(x)p_Y(y)})$$

$$= \sum_{x \in X} \sum_{y \in Y} p_{X,Y}(x,y)\ log\ log \left(\frac{p_{(X,Y)}(x,y)}{p_X(x)} \right) - \sum_{x \in X} \sum_{y \in Y} p_{X,Y}(x,y)$$
$$log\ log\ (p_Y(y))$$

$$= -\sum_{x \in X} p(x)H(X = x) - \sum_{y \in Y} p_Y(y) log(p_Y(y))$$

$$= -H(X) + H(Y)$$

$$= H(Y) - H(Y|X)\ (*)$$

Trong đó : $H(X)$ là entropy X

Theo trực quan:

+ Entropy $H(Y)$ được coi là thước đo mức độ đảm bảo của một biến ngẫu nhiên.
+ $H(Y|X)$ là thước đo của lượng thông tin X không nói về Y. Đây là "mức độ đảm bảo về Y sau khi X được biết"
+ Và do đó, (*) có thể được đọc là "mức độ chắc chắn trong Y, trừ đi mức độ chắc chắn trong Y khi biết X"
+ Tương đương với "độ chắc chắn trong Y khi được loại bỏ thông tin về X".
+ Điều này chứng thực ý nghĩa trực quan của tương trợ là lượng thông tin (nghĩa là giảm độ không chắc chắn) mà việc biết một trong hai biến cung cấp về biến kia.

6. INCEPTION SCORE

Inception Score (IS) **[3]** là một thang đo để tự động đánh giá chất lượng của các mô hình tạo hình ảnh.

IS sử dụng một mạng học sâu được huấn luyện trước trên tập dữ liệu cho trước với kích thước ***N*** và tính toán thống kê kết quả đầu ra của mạng khi áp dụng cho các hình ảnh được sinh ra bởi Generator. Công thức IS được cho bởi:

$$IS(G) = exp\ (E_{x \sim p_g} D_{KL}(p(y|x)||p(y))$$

Trong đó:

- $x \sim p_g$ nghĩa là x là một mẫu dữ liệu ảnh được từ p_g
- $D_{KL}(p(y|x)$ là độ đo Kullback-Leibler divergence **[14]** từ y vào x
- $p(x)$ là phân phối có điều kiện
- $p(y) = \int_x p(x) p_g x$ là phân phối cận biên (phân phối lề)

Hàm exp nhằm giúp hỗ trợ tính toán dễ dàng hơn. Chúng ta có thể bỏ qua bằng cách sử dụng $ln\ ln\ (IS(G)$.

Phương pháp IS nhằm hệ thống hóa hai tính chất mong muốn của một mô hình chung thành một thước đo:

- Hình ảnh được tạo ra phải chứa các vật thể rõ ràng (tức là hình ảnh sắc nét chứ không phải mờ) hoặc $p(x)$ phải có entropy thấp. Nói cách khác, mạng khởi động nên tin chắc rằng có một đối tượng duy nhất trong hình ảnh.
- Thuật toán tổng hợp phải tạo ra độ đa dạng cao của hình ảnh từ tất cả các lớp khác nhau trong mạng học sâu được huấn luyện ban đầu, tức là $p(y)$ phải có entropy cao.

Nếu cả hai đặc điểm này đều được thỏa mãn bởi một mô hình tổng hợp, thì chúng ta mong đợi Kullback-Leibler divergence lớn giữa các phân phối p (y) và p (y | x), dẫn đến IS lớn.

Ta thực hiện biến đổi và đánh giá như sau:

1. Biến đổi trên biểu thức đánh giá :

$$ln\ ln\ (IS(G)\ ) = E_{x\sim p_g} D_{KL}(p(y|x)||p(y))$$

$$= \sum_x \quad p_g(x) D_{KL}(p(x)\ ||p(y))$$

$$= \sum_x \quad p_g(x) \sum_i \quad p(x)\ ln \frac{p(x)}{p(y=i)}$$

$$= \sum_x \quad \sum_i \quad p(x, y=i) ln \frac{p(y=i)}{p(x)p(y=i)} = I(y;x) = H(y) - H(y|x)$$

(Theo tính chất 3 của tương trợ thông tin)

Điều này có nghĩa là :

Ta có : $I(y,x) \geq 0\ nên\ H(y) - H(x) \geq 0, suy\ ra$:

$$0\ \leq H(y) - H(x) \leq H(y) \leq ln\ ln\ (N)$$

N là kích thước của tập dữ liệu ảnh cho trước (ảnh thật)

Do đó : $1 \leq\ IS(G) \leq N$ và để IS(G) đạt được giá trị cực đại thì :

+ $H(y) = ln\ (N)$ khi và chỉ khi $p(y)$ là một phân phối chuẩn

+ $H(x) =\ 0$ xảy ra khi $p(x) = 1$ với duy nhất một i và các $j\ \neq i$ thì $p(x) = 0$

2. Thực hiện phương pháp đánh giá:

Với các mẫu $x^{(i)}$. Ta trước tiên tính phân phối lề :

$$\hat{\rho}(y) = \frac{1}{m}\sum_{i=1}^{m} \quad p(x^{(i)})$$

$$IS(G) \approx exp\ (\frac{1}{m}\sum_{i=1}^{m} \quad D_{KL}(p\left(x^{(i)}||\hat{\rho}(y)\right)$$

7. THỰC NGHIỆM VÀ KẾT QUẢ

7.1. Thực nghiệm

7.1.1. Dữ liệu

Về phần dữ liệu, chúng tôi thực hiện huấn luyến trên tập dữ liệu Fashion MNIST**[4]** với 10 lớp đối tượng, mỗi lớp với 6000 ảnh trong tập huấn luyện và 1000 ảnh trong tập kiểm tra. Ở thực nghiệm với GAN, chúng tôi chỉ sử dụng tập kiểm tra với 60000 ảnh với kích thước mỗi ảnh là $28 \times 28 \times 1$. Bên cạnh đó chúng tôi của thực hiện thử nghiệm trên tập dữ liệu ANIME **[10]** để xem xét sự khác biệt khi thực hiện huấn luyện ảnh màu.

7.1.2. Chi tiết cài đặt

Chi tiết phần cài đặt cho cả mô hình GAN và DCGAN, chúng tôi huấn luyện với 200 epochs và tốc độ học (learning rate) là $1e-4$. Thuật toán tối ưu sử dụng là Adam và đánh giá chi phí mỗi bước huấn luyện thông qua Binary Cross Entropy Loss.

Với kích cỡ chiều khởi tạo của không gian nhiễu, chúng tôi mặc định là 100. Kích thước ảnh giả được tạo ra bởi module Generator sẽ là $28 \times 28 \times h$, với h là số kênh màu tương ứng $h = 1$ khi sử dụng GAN và $h = 3$ khi sử dụng DCGAN.

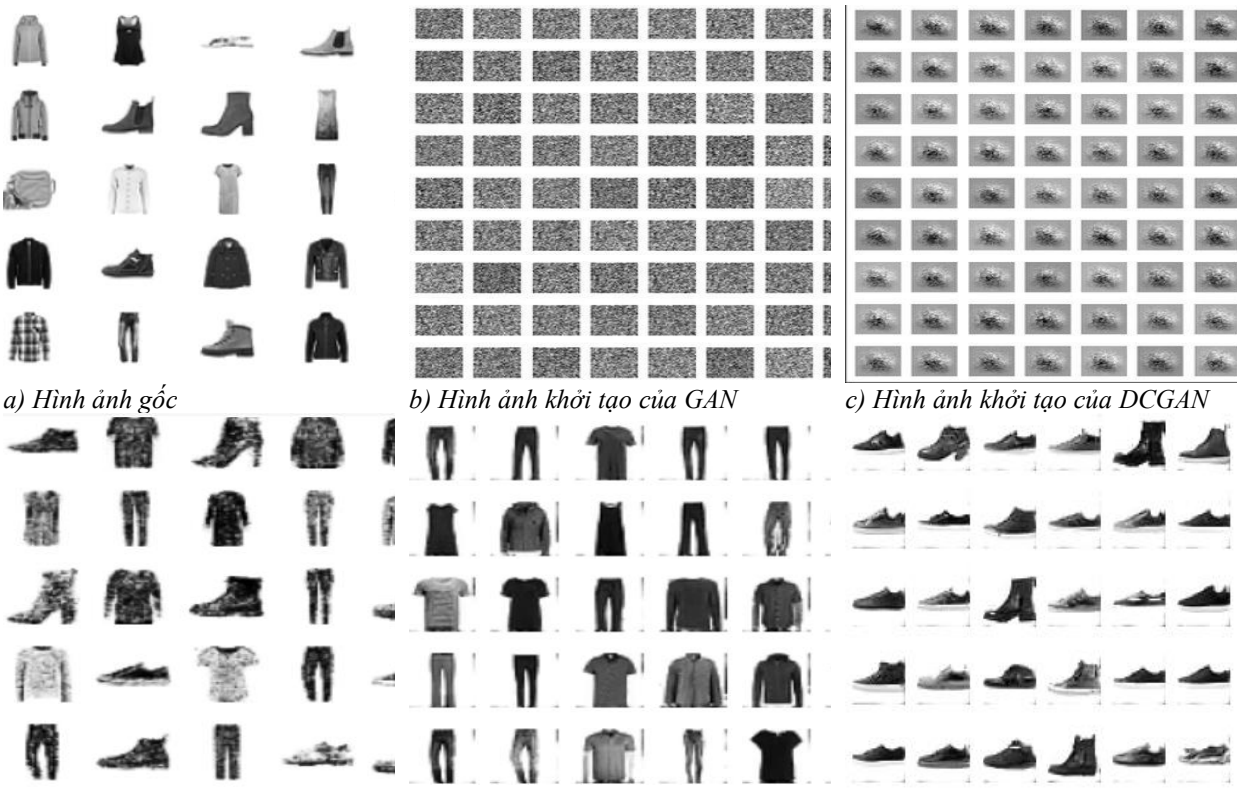
7.2. Kết quả

7.2.1. Kết quả thực nghiệm

Với mô hình GAN, sau quá trình huấn luyện chúng tôi sinh được một tập dữ liệu giả, với sự quan sát định tính, chúng tôi đánh giá rằng dữ liệu được sinh ra còn kém chất lượng và chưa có sự đa dạng nhất định.

Với mô hình DCGAN, với lần huấn luyện đầu tiên chúng tôi nghĩ rằng mô hình của mình đã gặp phải hiện tượng mode collapse khi trong 10 epochs đầu, module Generator chỉ sinh ra toàn ảnh cùng một đối tượng (cụ thể là hình ảnh giày) (Hình vẽ). Vì thế, chúng tôi đã phải thực hiện huấn luyện lại lần hai và thu được dữ liệu sinh ra đa dạng và có chất lượng tốt hơn.

Bên cạnh đó, chúng tôi cũng thực hiện huấn luyện mô hình DCGAN trên tập dữ liệu ANIME để quan sát kết quả.



a) Hình ảnh gốc *b) Hình ảnh khởi tạo của GAN* *c) Hình ảnh khởi tạo của DCGAN*

d) Hình ảnh sinh ra bởi GAN *e) Hình ảnh sinh ra bởi DCGAN* *f) Hình ảnh hiện tượng mode collapse bởi DCGAN*

Hình 3: Một số hình ảnh dữ liệu sinh ra trong quá trình huấn luyện

a) Hình ảnh gốc *b) Hình ảnh sinh ra bởi DCGAN*

Hình 4: *Kết quả mô hình DCGAN trên bộ dữ liệu ANIME*

7.2.2. Kết quả đánh giá

Ở bước đánh giá, chúng tôi chọn ngẫu nhiên 1000 ảnh trên mỗi tập dữ liệu *(bao gồm cả dữ liệu gốc*) và thực hiện đánh giá bằng phương pháp Inception Score. Nhận được kết quả tốt nhất với dữ liệu được sinh ra bởi DCGAN *(không bao gồm dữ liệu gốc).*

Dataset (Fashion MNIST)	**Inception Score**	
	AVG	**STD**
Original data	*4.6558*	*0.2116*
Generative by GAN model	*3.1660*	*0.1673*
Generative by DCGAN model	***3.4590***	***0.1284***
Generator by DCGAN model (Mode collapse)	*2.0260*	*0.0895*

Bảng 1: Kết quả đánh giá mô hình GAN và DCGAN trên tập dữ liệu Fashion MNIST

8. KẾT LUẬN

Qua bài báo cáo này, chúng tôi đã thực hiện tìm hiểu và phân tích các mặt toán học của thuật toán Generative Adversarial Networks cùng phương pháp đánh giá Inception Score. Chúng tôi cũng đã thực hiện xây dựng thành công hai mô hình GAN và DCGAN và thực hiện đánh giá trên tập dữ liệu Fashion MNIST.

Thông qua việc thử nghiệm, tìm hiểu và nghiên cứu này đã giúp nhóm chúng tôi hiểu sâu hơn các vấn đề gặp phải của mô hình GAN. Trong tương lai, chúng tôi vẫn sẽ dự định tiếp tục nghiên cứu đề tài này sâu hơn bằng cách tìm hiểu các biến thể của mô hình GAN và các phương pháp giải quyết nhược điểm của mô hình GAN.

9. TÀI LIỆU THAM KHẢO

[1] Generative Adversarial Networks, Ian J.Goodfelow, arXiv:1406.2661, 2014.

[2] Unsupervised Representation Learning with Deep Convolutional Generative. Adversarial Networks, Alec Radford,ICLR , arXiv:1511.06434, 2016.

[3] A Note on the Inception Score, Shane Barrat, arXiv:1801.01973, 2018.

[4] Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms, Han Xiao, arXiv:1708.07747, 2017.

[5] Convergence Problems with Generative Adversarial Networks, S. A. Barnett.

[6] Math behind gans, Mayank Vadsola Mayank Vadsola.

[7] Understanding latent space in machine learning, Ekin Tiu.

[8] Minimax – Wikipedia.

[9] Mutual information - Scholarpedia

[10] Anime Face Dataset | Kaggle

[11] John von Neumann – Wikipedia

[12] Latent space - Wikipedia

[13] Radon–Nikodym theorem - Wikipedia

[14] Kullback–Leibler divergence - Wikipedia

[15] Jensen–Shannon divergence - Wikipedia