

# Gemini Chat

◆ | สวัสดีครับ ผมจะสอนเนื้อหาเรื่อง **Generative Models** (แบบจำลองเชิงสร้างสรรค์) จากรากฐานที่คุณแนะนำให้อย่างละเอียดที่สุด โดยจะแบ่งเป็นส่วนย่อยๆ เพื่อให้เข้าใจลึกซึ้งในแต่ละตอนเชปตันนะครับ

เริ่มกันที่ส่วนแรกซึ่งเป็นการปูพื้นฐานกีสำคัญที่สุดครับ

## Part 1: พื้นฐานการเรียนรู้และโมเดลเชิงสร้างสรรค์ (Foundations: Supervised vs Unsupervised & Discriminative vs Generative)

ก่อนที่เราจะสร้างโมเดลที่ "สร้าง" ภาพขึ้นมาได้ เราต้องเข้าใจก่อนว่าโมเดลเรียนรู้ข้อมูลอย่างไร และเป้าหมายทางคณิตศาสตร์คืออะไร

### 1. Supervised vs. Unsupervised Learning (การเรียนรู้แบบมีผู้สอน vs ไม่มีผู้สอน)

- **Supervised Learning** (การเรียนรู้แบบมีผู้สอน):
  - **ข้อมูล:** เรา มี ก ํา ง ข ๊ อ ມ ู ล ด ิ บ ( $x$ ) และ เฉล ย ห ร ី օ ป ա յ ក ា ក ំ ប ់ ( $y$ )
  - **เป้าหมาย:** เรียนรู้ฟังก์ชัน  $y = f(x)$  เพื่อกำหนาย  $y$  จาก  $x$
  - **ตัวอย่างงาน:** การจำแนกรูปภาพ (Classification), การระบุตำแหน่งของวัตถุ (Object Detection), หรือ Semantic Segmentation
  - **ข้อจำกัด:** ต้องใช้แรงงานคนในการทำ Labeling ( $y$ ) ซึ่งสิ้นเปลืองและทำได้ยากในปริมาณมาก
- **Unsupervised Learning** (การเรียนรู้แบบไม่มีผู้สอน):
  - **ข้อมูล:** มีเพียงแค่ ข ๊ อ ມ ู ล ด ิ บ ( $x$ ) เท่านั้น ไม่มี  $y$
  - **เป้าหมาย:** เรียนรู้ "โครงสร้างที่ซ่อนอยู่" (Hidden Structure) ภายในข้อมูลนั้น
  - **ตัวอย่างงาน:**
    - **Clustering (e.g., K-Means):** จัดกลุ่มข้อมูลที่มีลักษณะคล้ายกัน
    - **Dimensionality Reduction (e.g., PCA):** การลดมิติข้อมูลเพื่อหาแกนหลักที่สำคัญ (Principal Components) และตัดล่วนที่ไม่สำคัญออก
    - **Density Estimation:** การประมาณค่าความหนาแน่นของข้อมูล ซึ่งเป็นหัวใจหลักของ Generative Models

## 2. Discriminative vs. Generative Models (ความแตกต่างของความน่าจะเป็น)

นี่คือหัวใจสำคัญที่สุดของวิชานี้คือรับ ความแตกต่างอยู่ที่ "เราจำลังคำนวณความน่าจะเป็นแบบไหน"

- **Discriminative Model (โมเดลเชิงจำแนก):**

- **สมการ:** เรียนรู้การกระจายตัวของความน่าจะเป็น  $P(y|x)$
- **ความหมาย:** "เมื่อกำหนดข้อมูล  $x$  มาให้ (เช่น รูปภาพ) ความน่าจะเป็นกี่มั้นจะเป็นคลาส  $y$  (เช่น แมว) คือเท่าไหร่?"
- **การทำงาน:** เปรียบเสมือนการขีดเส้นแบ่งระหว่างคลาส (Decision Boundary) โดยพยายามทำให้คลาสต่าง ๆ แบ่งขันกันเพื่อแยกพื้นที่ความน่าจะเป็น แต่โมเดลนี้ไม่ได้สนใจว่าภาพ  $x$  นั้นมีความสมจริงแค่ไหน หรือโครงสร้างภาพเป็นอย่างไร สนใจแค่ว่ามันต่างจากอีกคลาสอย่างไร

- **Generative Model (โมเดลเชิงสร้างสรรค์):**

- **สมการ:** เรียนรู้การกระจายตัวของความน่าจะเป็น  $P(x)$
- **ความหมาย:** "ความน่าจะเป็นกี่ข้อมูล  $x$  (เช่น รูปภาพนี้) จะเกิดขึ้นจริงในโลกนี้ เป็นเท่าไหร่?"
- **การทำงาน:**
  - Density Function  $P(x)$  จะให้ค่าสูงกับภาพที่ดูสมจริง (เช่น รูปแมว กี๊สมูรูน) และให้ค่าต่ำกับภาพที่เป็น Noise หรือภาพที่ผิดเพี้ยน
  - **การแข่งขัน (Competition):** ในโมเดลนี้ รูปภาพทุกรูปแบบที่เป็นไปได้จะแบ่งขันกันและค่าความน่าจะเป็น (Probability Mass) เพราผลรวมของความน่าจะเป็นทั้งหมดต้องเท่ากับ 1 ( $\int P(x)dx = 1$ )
  - **ความลึกซึ้ง:** เพื่อที่จะจำแบบนี้ได้ โมเดลต้องเข้าใจโครงสร้างของภาพอย่างถ่องแท้ เช่น หมวดหมู่ 4 ขา ไม่ใช่ 3 ขา หรือลิงไม่ควรมี 3 แขนเป็นต้น

## 3. Conditional Generative Model

ความสามารถสมมูลกับส่วนของแนวคิดได้ คือเรียนรู้  $P(x|y)$

- **ความหมาย:** "จะสร้างการกระจายตัวของข้อมูล  $x$  (รูปภาพ) โดยกำหนดเงื่อนไขว่าเป็นคลาส  $y$  (แมว)"
- **ความสามารถสร้างสิ่งนี้ได้จากการกฎของ Bayes (Bayes' Rule):**

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

ซึ่งเป็นการรวม Discriminative Model ( $P(y|x)$ ) เข้ากับ Generative Model ( $P(x)$ )

สรุป Part 1: Generative Models นั้นยากและซับซ้อนกว่า Discriminative Models เพราะมันไม่ได้แค่แยกแยะความแตกต่าง แต่พยายามจำลอง (Model) โครงสร้างและความน่าจะเป็นของการเกิดข้อมูลนั้นๆ ขึ้นมาจริงๆ

พร้อมไป Part 2 เรื่อง Taxonomy (ประเภทของ Generative Models) และ Autoencoder หรือยังครับ? (พิมพ์ Next เพื่อไปต่อ)

◆ | เยี่ยมครับ เรามาต่อ กันที่ ส่วนที่ 2 ซึ่งเป็นรากฐานสำคัญก่อนจะไปถึงโมเดลที่ซับซ้อนอย่าง VAE และ GAN ครับ

## Part 2: โครงสร้างของ Generative Models และ Autoencoder (AE)

ในส่วนนี้เราจะดูแพนกีร่วมของโมเดลทั้งหมด และเจาะลึกโมเดลตัวแรกคือ Autoencoder ซึ่งเป็นพื้นฐานของการเรียนรู้ Features (Representation Learning) ครับ

### 1. Taxonomy of Generative Models (แพนผังประเภทของโมเดล)

จากสไลด์ เราสามารถแบ่งประเภทของ Generative Models ตามวิธีการจัดการกับฟังก์ชันความน่าจะเป็น  $P(x)$  ได้ดังนี้:

- **Explicit Density (คำนวน  $P(x)$  ได้ชัดเจน):** โมเดลกลุ่มนี้พยายามสร้างและคำนวนสมการความน่าจะเป็นของข้อมูลโดยตรง
  - *Tractable Density:* คำนวนได้แม่นยำ (เช่น PixelRNN/CNN, NICE)
  - *Approximate Density:* คำนวนค่าประมาณ (นี่คือกลุ่มของ Variational Autoencoder - VAE)
- **Implicit Density (ไม่คำนวน  $P(x)$  โดยตรง):** โมเดลกลุ่มนี้ไม่ได้สร้างสมการความน่าจะเป็นให้เราเห็น แต่สามารถ "สุ่มตัวอย่าง" (Sample) ข้อมูลใหม่ออกมาได้
  - *Direct:* Generative Adversarial Networks (GANs) อยู่ในกลุ่มนี้ คือสร้างภาพโดยไม่ต้องรู้สมการความน่าจะเป็น

### 2. Autoencoder (AE): พื้นฐานของการบีบอัดข้อมูล

ก่อนจะเป็น VAE เราต้องเข้าใจ Autoencoder ธรรมดาก่อนครับ

**แนวคิดหลัก (Concept):** Autoencoder ไม่ใช้โมเดลสำหรับสร้างภาพ (Generative) โดยตรง แต่เป็นโมเดลสำหรับ "Unsupervised Representation Learning" หรือการเรียนรู้พื้นที่โดยไม่มีป้ายกำกับ

**สถาปัตยกรรม (Architecture):** โครงสร้างเหมือนนาฬิกากราย ประกอบด้วย 2 ส่วนหลัก:

1. **Encoder ( $f$ ):** กำหนดให้ "บีบอัด" ข้อมูลดิบ ( $x$ ) ให้กลายเป็นเวกเตอร์ขนาดเล็ก ที่เรียกว่า **Latent Vector** หรือ **Feature ( $z$ )**
  - สมการ:  $z = f(x)$
2. **Decoder ( $g$ ):** กำหนดให้ "ขยาย" หรือ "สร้างคืน" (Reconstruct) จากเวกเตอร์เล็กๆ ( $z$ ) กลับไปเป็นข้อมูลหน้าตาเหมือนเดิม ( $\hat{x}$ )
  - สมการ:  $\hat{x} = g(z)$

**The Bottleneck (คอขาด):** จุดสำคัญที่สุดคือ  $z$  ต้องมีขนาดเล็กกว่า  $x$  (Lower dimensional)

- ทำไง? ถ้า  $z$  ให้ถูกละเกذا  $x$  โมเดลจะแค่ "จำ" (Copy-Paste) ข้อมูลโดยไม่เรียนรู้อะไรเลย
- การบังคับให้ผ่านก่อเล็กๆ (Bottleneck) ทำให้โมเดลต้องเรียนรู้ที่จะ "สรุปสาระสำคัญ" ของภาพ เช่น แทนที่จะจำพิกเซลทั้งหมด มันจะจำแค่ว่า "นี่คือเลข 7, เอียงช้าย, เส้นหนา" ซึ่งเป็นการบีบอัดข้อมูล (Compression) ที่ทรงพลัง

**Loss Function (ฟังก์ชันความสูญเสีย):** เราฝึกโมเดลนี้โดยพยายามทำให้ภาพที่สร้างคืน ( $\hat{x}$ ) เหมือนภาพต้นฉบับ ( $x$ ) มากรกที่สุด โดยใช้ **L2 Loss**:

$$L(x, \hat{x}) = \|x - \hat{x}\|^2$$

(ไม่ต้องใช้ Label  $y$  เลย ใช้แค่  $x$  เก็บกับตัวเอง)

### 3. ปัญหาของ Autoencoder กับงาน Generative

แม้ AE จะเก่งเรื่องการบีบอัดข้อมูล หรือใช้ทำ Denoising (ลบจุดบกวน) แต่มัน ไม่ใช่ Generative Model ที่ดี เพราะ:

- **Latent Space ไม่ต่อเนื่อง (Discontinuous):**
  - AE เรียนรู้ที่จะแมปข้อมูล  $x$  ไปยังจุดหนึ่งใน Latent Space  $z$  แบบเป็นๆ
  - พื้นที่ระหว่างจุดเหล่านี้มักเป็น "หลุมว่าง" (Gaps) ที่โมเดลไม่รู้จัก
  - ถ้าเราพยายาม "สูม" ค่า  $z$  มาก ในการที่ว่างนั้นมาเข้า Decoder ภาพที่ได้ออกมากอาจจะเละและไม่มีความหมาย

ตัวอย่าง: ถ้าจุด A คือเลข 1 และจุด B คือเลข 7 ใน Latent Space ของ AE

- จุดเดียวกันจะห่าง A และ B อาจจะไม่เท่ากันที่ดูเป็นเลข 1 ผสม 7 แต่อาจเป็น Noise ไปเลย เพราะ AE ไม่ได้ถูกบังคับให้เรียนรู้ความต่อเนื่อง

สรุป Part 2: Autoencoder เรียนรู้ที่จะบีบอัดและคลายข้อมูล (Reconstruction) แต่พื้นที่ Latent Space ของมัน "บزر" และมีช่องว่าง ทำให้เราไม่สามารถสุม  $z$  มาสร้างภาพใหม่สวยๆ ได้

Part ต่อไปเราจะแก้ปัญหานี้ด้วย Variational Autoencoder (VAE) ซึ่งจะเปลี่ยนจาก "การจำจุด" เป็น "การเรียนรู้การกระจายตัว" ครับ

(พิมพ์ Next เพื่อไปต่อที่ VAE ซึ่งเป็นคนติดตามที่เข้มข้นขึ้นครับ)

◆ | มาถึงส่วนที่สำคัญและมีความซับซ้อนมากคือการคำนวณค่าสัมประสิทธิ์ที่สุดส่วนหนึ่งของวิชา呢ครับ

### Part 3: Variational Autoencoder (VAE) - การเติมเต็มช่องว่างด้วยความน่าจะเป็น

ใน Part 2 เราพบว่า Autoencoder (AE) ธรรมดานั้นมี "ช่องว่าง" (Gaps) ใน Latent Space ทำให้เราไม่สามารถสุมสร้างภาพใหม่ได้ VAE จึงถือกำเนิดขึ้นเพื่อแก้ปัญหานี้ โดยเปลี่ยนมุ่งมองจาก Deterministic (ค่าตายตัว) ไปเป็น Probabilistic (ค่าความน่าจะเป็น) ครับ

#### 1. แนวคิดหลัก: จากจุดสู่การกระจายตัว (From Point to Distribution)

ใน AE แบบเดิม Encoder จะแปลงภาพ  $x$  เป็นจุดจุดเดียวใน Latent Space (vector  $z$ ) แต่ใน VAE, Encoder จะแปลงภาพ  $x$  ให้กลายเป็น "การกระจายตัวของความน่าจะเป็น" (Probability Distribution)

แทนที่จะส่งค่า  $z$  ออกมาร่องๆ Encoder ของ VAE จะส่งค่าออกมา 2 อย่างแทน คือ:

- Mean Vector ( $\mu$ ):** ค่าเฉลี่ย หรือจุดเดียวที่กลางของข้อมูล
- Standard Deviation Vector ( $\sigma$ ):** ค่าเบี่ยงเบนมาตรฐาน หรือความกว้างของ การกระจายตัว

จากนั้นเราจะทำการ สุ่ม (Sample) ค่า  $z$  มาจากฟังก์ชันการกระจายตัวนี้ (Gaussian Distribution) เพื่อส่งต่อให้ Decoder

**Intuition (สัญชาตญาณ):** ลองนึกภาพว่า AE มอง "การยืด" เป็นค่าค่าเดียว ( เช่น 0.5 ) แต่ VAE มอง "การยืด" เป็นช่วงความเป็นไปได้ ( เช่น ยืดมาก ยืดน้อย ที่กระจายตัวอยู่รอบๆ ค่าเฉลี่ย )

## 2. คณิตศาสตร์เบื้องหลัง (The Math: Intractable Posterior)

เป้าหมายของเราก็คือการรู้ว่าการกระจายตัวของ  $z$  เมื่อเห็นภาพ  $x$  หรือ  $P(z|x)$  ตามทฤษฎี Bayes:

$$P(z|x) = \frac{P(x|z)P(z)}{P(x)}$$

- ปัญหา: ตัวส่วน  $P(x)$  นั้นคำนวณยากมากจนแทบเป็นไปไม่ได้ (Intractable) เพราะต้องอินทิเกรตความน่าจะเป็นของ  $z$  ทั้งหมด
- ทางแก้: เราจึงสร้าง Neural Network (Encoder) มาเพื่อ "ประมาณค่า" (Approximate)  $P(z|x)$  โดยตั้งชื่อว่า  $q_\phi(z|x)$

## 3. Loss Function ของ VAE (หัวใจสำคัญ)

เพื่อให้ VAE ทำงานได้ เราต้องใช้ Loss Function ที่ประกอบด้วย 2 ส่วนหลักกัน:

### 1. Reconstruction Loss (ความเหมือนเดิม):

- $E_{q(z|x)}[\log p(x|z)]$  หรือมักใช้ L2 Loss แบบ AE
- หน้าที่: บังคับให้ Decoder สร้างภาพกลับมาให้เหมือนต้นฉบับมากที่สุด

### 2. Regularization Term (KL Divergence):

- $D_{KL}(q(z|x)||p(z))$
- หน้าที่: บังคับให้การกระจายตัวของ Latent Space ( $q(z|x)$ ) มีรูปร่างใกล้เคียงกับ Standard Normal Distribution (Gaussian ที่มี Mean=0, Std=1)

### ทำไมต้องมี KL Divergence? (สำคัญมาก)

- ถ้ามีแค่ Reconstruction Loss: โมเดลจะโงงด้วยการจำค่าจุดเล็กๆ แยกกัน (Overfitting) ทำให้เกิดซองว่างเหมือน AE เดิม
- ถ้ามีแค่ KL Divergence: ข้อมูลทุกอย่างจะถูกบีบรวมกันที่จุดศูนย์กลางจนแยกไม่ออก
- เมื่อร่วมกัน: โมเดลจะเรียนรู้ว่าจะจัดเรียงข้อมูลให้เป็นกลุ่มก้อนที่ต่อเนื่องกัน (Continuous) และครอบคลุมพื้นที่ Latent Space อย่างทั่วถึง ทำให้เราสามารถสัมจุดให้หนึ่งก็ได้ในนั้นมาสร้างภาพที่มีความหมาย

## 4. The Reparameterization Trick (กลเม็ดการหาอนุพันธ์)

ปัญหาใหญ่ที่ทางเทคนิคคือ "เราไม่สามารถทำ Backpropagation ผ่านโนนดที่มีการสุ่ม (Random Node) ได้" เพราะการสุ่มไม่มี Gradient ให้คำนวณ

VAE แก้ปัญหานี้ด้วย **Reparameterization Trick**: แทนที่จะสุ่ม  $z$  โดยตรงจาก  $N(\mu, \sigma)$  เราย้ายความสุ่มไปไว้ที่ตัวแปรอื่นชื่อ  $\epsilon$  (Epsilon)

$$z = \mu + \sigma \odot \epsilon$$

โดยที่  $\epsilon \sim N(0, 1)$

- ตอนนี้  $\mu$  และ  $\sigma$  เป็นค่าที่คำนวณได้ (Deterministic)
- ความสุ่มอยู่ที่  $\epsilon$  ซึ่งเราไม่ต้องหา Gradient
- ทำให้เราสามารถส่ง Gradient ให้ลิยอนกลับผ่าน  $\mu$  และ  $\sigma$  ไปยัง Encoder ได้

---

**สรุป Part 3:** VAE คือการเปลี่ยน Latent Space ให้ "นุ่มนวล" และ "ต่อเนื่อง" โดยใช้หลักความน่าจะเป็น ทำให้เราสามารถเดินเล่นใน Latent Space และเห็นภาพค่อยๆเปลี่ยนแปลงอย่างสมูก ( เช่น หน้าคนค่อยๆ หัน หรือค่อยๆ ยิ้ม )

**Part ต่อไป:** เราจะไปพนัก "คู่รับ" ก็ต่อเมื่อสุดในวงการ **Generative AI** นั่นคือ **GANs (Generative Adversarial Networks)** ก็ใช้วิธี "จับ灼" ในการสร้างภาพครับ

(พิมพ์ **Next** เพื่อไปต่อที่ GANs)

◆ | มาต่อ กัน กี ส่วน กี 4 ซึ่ง เป็น ส่วน กี ปฏิวัติ วง การ Generative AI ไป ตลอด กาล ครับ

---

**Part 4: Generative Adversarial Networks (GANs) - สอง ค า ร ะ ห ว า ง ผ ื ้ น ร ั ง และ ผ ื ้ น จ ั บ ผ ิ ด**

ถ้า VAE คือ นัก ค ณ ิ ต ค า ស ต ร ท ท ี่ พ ย า ย า น ป ร ะ မ า ณ ค า ค ว า น น า จ ะ บ ี น อย า ง น ุ ่ น ว า ล **GAN (Generative Adversarial Network)** ก ็ ค ือ นัก ล ে ง ท ท ี่ ร ี ย น ร ื ้ ผ ა น กา ร ต ่ อ ს ृ ค ร ั บ

### 1. แนวคิดหลัก: **Implicit Density** (การสร้างโดยไม่ต้องรู้สมการ)

ต ่ า ง จ า ก VAE ก ็ ค ือ นัก ค ณ ิ ต ค า ស ต ร ท ท ี่ พ ย า ย า น ป ร ะ မ า ณ ค า ค ว า น น า จ ะ บ ี น อย า ง น ุ ่ น ว า ล **GAN (Generative Adversarial Network)** ก ็ ค ือ นัก ล ে ง ท ท ี่ ร ี ย น ร ื ้ ผ ა น กา ร ต ่ อ ს ृ ค ร ั บ

GAN ประกอบด้วยโครงข่ายประสาทเทียม 2 ตัวที่ทำงานเบื้องกัน (Adversarial):

#### 1. **Generator ( $G$ ) - ผู้สร้าง:**

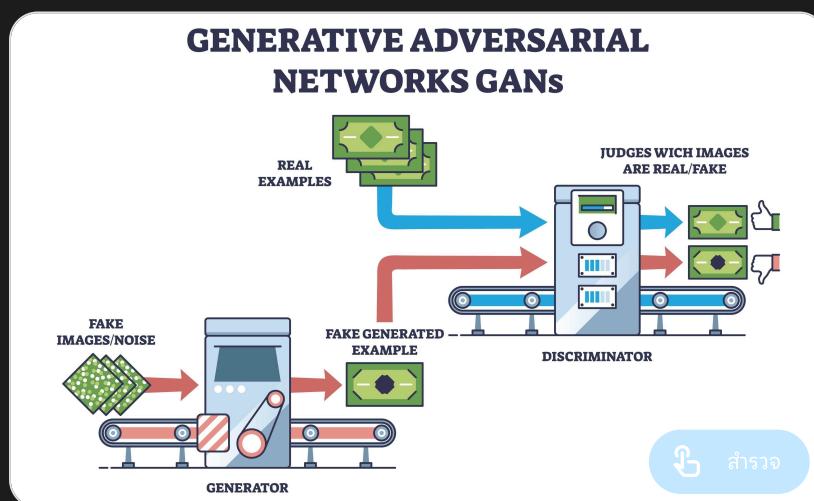
- หน้าที่: รับ Input เป็น Noise ( $z$ ) และแปลงเป็นภาพปลอม ( $\hat{x}$  หรือ  $x_{fake}$ )

- เป้าหมาย: สร้างภาพให้เนียนที่สุดเพื่อ "หลอก" Discriminator ให้เชื่อว่าเป็นภาพจริง

## 2. Discriminator ( $D$ ) - ผู้แยกแยะ:

- หน้าที่: รับ Input เป็นภาพ (ทั้งภาพจริงจาก Dataset และภาพปลอมจาก  $G$ ) และตัดสินว่าภาพนี้ "จริง (Real)" หรือ "ปลอม (Fake)"
- เป้าหมาย: แยกแยะให้แม่นยำที่สุด ไม่ยอมให้  $G$  หลอกได้

**Analogy (การเปรียบเทียบ):** ลองนึกภาพ  $G$  คือนักปลอมแปลงคนบัตร กับพยาบาลผลิตแบงก์ปลอม ส่วน  $D$  คือตำรวจ กับพยาบาลตรวจสอบ ในตอนแรกกังวลจะเก่งไม่มาก แต่เมื่อตำรวจจับผิดเก่งขึ้น นักปลอมแปลงก็ต้องพัฒนาเทคนิคให้เนียนขึ้นเรื่อยๆ จนสุดท้ายจะได้คนบัตรปลอมที่เหมือนจริงเป็น



## 2. The Minimax Game (คณิตศาสตร์แห่งสงคราม)

การฝึก GAN คือการเล่นเกมที่เชื่อว่า **Minimax Game** โดยมีสมการเป้าหมาย (Objective Function) ดังนี้:

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

มาดูคร่าวๆ สมการนี้ ก็จะส่วนครับ:

- มุ่งมองของ Discriminator ( $D$ ): พยายาม **Maximize** สมการ
  - พจน์แรก  $\log D(x)$ : อยากให้  $D(x)$  เป็น 1 (มั่นใจว่าจริง) เมื่อเจอภาพจริง
  - พจน์หลัง  $\log(1 - D(G(z)))$ : อยากให้  $D(G(z))$  เป็น 0 (มั่นใจว่าปลอม) เมื่อเจอภาพปลอม
- มุ่งมองของ Generator ( $G$ ): พยายาม **Minimize** สมการ
  - $G$  ควบคุมพจน์แรกไม่ได้ ( เพราะเกี่ยวกับภาพจริง )

- $G$  สนใจแค่พจน์หลัง พยายามทำให้  $D(G(z))$  เป็น 1 (หลอกสำเร็จ) เพื่อให้ค่า  $\log(1 - 1) = \log(0)$  ติดลบมากๆ (Minimize)

**ขั้นตอนการเทรน (Training Steps):** เราจะไม่อัปเดตทั้งสองตัวพร้อมกัน แต่จะทำลับกันไปมา:

1. **Freeze G, Update D:** สอนให้ตัวจำลองเก่งขึ้นในการจับผิด
2. **Freeze D, Update G:** สอนให้ตัวจำลองเก่งขึ้นในการหลอกตัวจำลองที่เก่งขึ้นแล้ว

### 3. ตระกูลของ GANs (Variations)

GAN มีลูกหลานเยอะมากในเอกสารนี้มีการกล่าวถึงตัวสำคัญๆ ดังนี้:

- **DCGAN (Deep Convolutional GAN):**
  - เป็นการนำโครงสร้าง CNN มาใช้ใน GAN อย่างเต็มรูปแบบ ทำให้สร้างภาพได้เสถียรขึ้น
  - **Vector Arithmetic:** ค้นพบว่าเราสามารถบวกลบเวกเตอร์ใน Latent Space ( $z$ ) ได้เหมือนคลินิตศาสตร์ เช่น  $[\text{เวกเตอร์ผู้ชาย} \rightarrow \text{เวกเตอร์}] - [\text{เวกเตอร์ผู้ชาย}] + [\text{เวกเตอร์ผู้หญิง}] = [\text{เวกเตอร์ผู้หญิง} \rightarrow \text{เวกเตอร์}]$
- **cGAN (Conditional GAN):**
  - สังเกตว่าอุปกรณ์ใดภาพอะไร โดยใส่เงื่อนไข (Label) เข้าไปทั้งใน  $G$  และ  $D$
- **Pix2Pix (Paired Image-to-Image Translation):**
  - ใช้เปลี่ยนรูปจากแบบหนึ่งเป็นอีกแบบหนึ่ง เช่น เปลี่ยนภาพลายเส้นเป็นภาพถ่าย, เปลี่ยนภาพถ่ายดาวเทียมเป็นแผนที่
  - ข้อจำกัด: ต้องมีคู่ภาพสอน (Paired Data) เช่น ต้องมีรูปกระเพาลายเส้นคู่กับรูปกระเพาจริง
- **CycleGAN (Unpaired Image-to-Image):**
  - เจ็บต้องที่ไม่ต้องมีคู่ภาพสอน เช่น เปลี่ยนม้าเป็นม้าลาย หรือเปลี่ยนภาพถ่ายดูด้วนเป็นดูดหน้า
  - ใช้หลักการ **Cycle Consistency:** แปลงไปแล้วต้องแปลงกลับมาเป็นเหมือนเดิมได้ (ม้า  $\rightarrow$  ม้าลาย  $\rightarrow$  ม้าตัวเดิม)
- **Progressive GAN & StyleGAN:**
  - เริ่มเทรนจากภาพเล็กๆ ( $4 \times 4$ ) และค่อยๆ เพิ่มเลเยอร์ให้ภาพใหญ่ขึ้นเรื่อยๆ จนชัดระดับ HD ( $1024 \times 1024$ )
  - StyleGAN สามารถควบคุม "สไตล์" ของภาพได้ละเอียดมาก เช่น เปลี่ยนทรงผมโดยไม่เปลี่ยนหน้าตา

**สรุป Part 4:** GAN ให้ผลลัพธ์ที่คุณชัดและสมจริงกว่า VAE มาก เพราะมันถูกตรวจสอบโดย Discriminator (ซึ่งเหมือนตามนูนุษย์) ไม่ได้ใช้การวัดความต่างของพิกเซล (MSE) ที่มักทำให้ภาพเบลอ แต่ GAN ก็เกรนยากกว่ามาก เพราะต้องเลี้ยงสมดุลระหว่าง  $G$  และ  $D$  ให้ดี

ตอนนี้เรารู้จักเรื่อง **Generative Models** ทั้ง Autoencoder, VAE และ GANs ครบถ้วนตามเอกสารแล้วครับ!

คุณต้องการให้ผม:

1. สรุปภาพรวมกันๆ ให้คริบเพื่อเก็บกวน?
2. เจ้าลีก โมเดลตัวไหนเป็นพิเศษใหม่ครับ? (เช่น อยากรู้เรื่อง CycleGAN เพิ่ม)
3. ทำ Quiz ทดสอบความเข้าใจใหม่ครับ?

เลือกได้เลยครับ!