Generative Adversarial Network (GAN)

Course:
Deep Learning with Tensorflow & Kersa 2

FANSHAWE

Developed by:
Mohammad Noorchenarboo

November 17, 2025

Current Section

1 What Is a Generative Adversarial Network?

2 Probabilities and Real vs. Fake Decisions

3 Understanding the GAN Loss Function

The Basic Idea of GANs

Motivating Question: Can a machine learn to create new and realistic content–like art or photos–without being explicitly told the rules of creation?

Terminology: GAN

A **Generative Adversarial Network (GAN)** is a machine learning system made up of two parts:

- A **generator**: its job is to create data that *looks* real.
- A **discriminator**: its job is to detect whether data is real or fake.

They are called *adversarial* because they are trained in competition with each other.

Core Concept

GANs aim to produce new data that mimics the structure, style, or essence of real-world data. They do this by learning through feedback–without explicitly modeling probability or rules.

Understanding Through Analogy

The Painter and the Art Critic

Imagine a young artist (the generator) learning to paint like a master.

- The artist creates paintings and shows them to an art critic (the discriminator).
- The critic evaluates each painting and decides whether it's an authentic masterpiece or an imitation.
- The artist improves their style by observing the critic's feedback.
- The critic sharpens their eye to detect more subtle flaws.

Over time, the artist becomes so skilled that the critic can no longer tell the difference.

Why Are GANs Important?

Real-World Use Cases

- **Image Generation:** Creating realistic human portraits or fantasy art.
- **Data Augmentation:** Generating more training data from limited samples.
- **Super Resolution:** Enhancing blurry images into sharper ones.
- **Creative AI:** Style transfer, synthetic music, and AI-generated storytelling.

Caveat

GANs generate data that *appears* real, but they do not understand the content they produce. Their learning is based on mimicry, not comprehension.

Summary: Conceptual Foundation of GANs

Concept	Explanation
Generator	A model that tries to produce new data that “looks real” to the discriminator.
Discriminator	A model that evaluates data and tries to guess whether it’s real or generated.
Adversarial Training	A competitive game where the generator and discriminator improve by challenging each other.
Realistic Data Generation	The ability to synthesize new, believable data from scratch.

Current Section

1 What Is a Generative Adversarial Network?

2 Probabilities and Real vs. Fake Decisions

3 Understanding the GAN Loss Function

Why Probability?

Motivating Question: If a model outputs "this image is real with 93% confidence", what does that really mean? And how do we represent this kind of binary certainty?

Bernoulli Distribution

The **Bernoulli distribution** is a discrete distribution with only two possible outcomes:

$$x \sim \mathrm{Bern}(p) \quad \Rightarrow \quad x \in \{0, 1\}, \text{ where } \begin{cases} P(x = 1) = p \\ P(x = 0) = 1 - p \end{cases}$$

Used to model binary events: real (1) vs. fake (0) in GANs.

Why Probability?

Example: Real or Fake Image

A discriminator predicts an image as real with $p = 0.93$:

$$P(x = 1) = 0.93, \quad P(x = 0) = 0.07$$

This is modeled as a Bernoulli trial: one sample, two outcomes.

What Is Log Probability?

Log Probability

The **log-probability** of an event is simply the logarithm of its likelihood:

$$\log P(x)$$

Why log? Because:

- Logs turn products into sums (useful for multiple observations).
- It penalizes wrong predictions more heavily when confidence is high.
- Common in statistical modeling and inference.

What Is Log Probability?

Example: Discriminator Predicts Image Is Real

If the discriminator predicts an image is real with $P = 0.9$, then:

$$\log P(x = 1) = \log(0.9) \approx -0.105 \text{ (high confidence)}$$

But if it predicted with low confidence, say $P = 0.1$:

$$\log P(x = 1) = \log(0.1) \approx -2.3 \text{ (penalized more)}$$

Intuition Behind $\log p$ in Real vs. Fake

Truth Detector as a Betting Game

Imagine you are a judge (discriminator) at an art contest. Every time a painting is shown, you bet money on whether it's real.

- Saying "I'm 95% sure this is real" is like betting heavily on it being genuine.
- If it turns out fake, your loss is large – because you were confident and wrong.
- $\log p$ captures this: low log-probability for confident wrong guesses.

Caution: Confidence Without Accuracy

A high p value means the model is confident – not necessarily correct. $\log(p)$ is only meaningful when compared with the true label (real or fake).

Summary: Binary Probabilities and Log Measures

Concept	Explanation
Bernoulli Distribution	Models binary outcomes (e.g. real vs. fake) with probability p for 1, $1-p$ for 0.
Probability Score	The discriminator's output is a number in $[0, 1]$, interpreted as confidence that input is real.
Log-Probability	$\log P(x)$ assigns lower scores to confident wrong predictions and rewards correct ones.
Interpretation	In GANs, real/fake predictions are treated as probabilistic binary events.

Current Section

1 What Is a Generative Adversarial Network?

2 Probabilities and Real vs. Fake Decisions

3 Understanding the GAN Loss Function

Motivating the Loss Function in GANs

Motivating Question: If a GAN tries to generate data that "looks real", how do we quantify what "looking real" even means? What should the generator and discriminator actually optimize?

Core Idea of Adversarial Loss

The GAN loss is built on a game:

- The **discriminator** tries to assign high probability to real data and low probability to fake data.
- The **generator** tries to produce fake data that the discriminator believes is real.

This creates a two-player minimax game between the generator G and the discriminator D.

Mathematical Formulation of the GAN Objective

Original GAN Objective (Goodfellow et al., 2014)

The minimax loss is defined as:

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

Where:

- $D(x)$: Discriminator's predicted probability that input x is real
- $G(z)$: Generator's output given random noise $z \sim p_z$

Interpretation: What Each Term Means

- $\log D(x)$: Rewarding the discriminator for correctly calling real samples "real"
- $\log(1 - D(G(z)))$: Rewarding the discriminator for calling fake samples "fake"
- The generator tries to *fool* the discriminator by maximizing $D(G(z))$

Why Use Log Probabilities?

Log Loss Is Derived from Bernoulli Likelihood

For binary classification with label $y \in \{0, 1\}$ and prediction $p = D(x)$, the log-likelihood is:

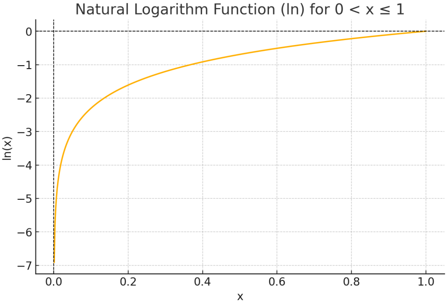
$$\log P(y \mid x) = y \log p + (1 - y) \log(1 - p)$$

This penalizes wrong predictions more heavily when confidence is high â consistent with how we want the discriminator to behave.

Grading a Judge's Confidence

A judge says: "I'm 99% sure this painting is real." If the painting turns out to be fake, the punishment should be large. That's what $\log(1 - D(G(z)))$ captures: confident wrong guesses are severely penalized.

The Generator's Goal: Fool the Discriminator



The Generator's Goal: Fool the Discriminator

Generator's Objective

While the discriminator tries to **maximize** classification accuracy, the generator tries to **minimize** the probability that its outputs are identified as fake.

$$\min_G \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))]$$

In practice, this can lead to vanishing gradients when $D(G(z)) \approx 0$. So an alternative (non-saturating) loss is often used:

$$\max_G \mathbb{E}_{z\sim p_z}[\log D(G(z))]$$

This has the same fixed point but better gradients early in training.

The Generator's Goal: Fool the Discriminator

Note

The non-saturating loss is not part of the original GAN paper but is widely used in practical implementations due to improved stability.

Summary: GAN Loss Function Essentials

Component	**Meaning**
$\log D(x)$	Discriminator reward for correctly identifying real data
$\log(1 - D(G(z)))$	Discriminator reward for catching fake data
Generator Loss	Pushes $D(G(z))$ closer to 1 â making fake data look real
Minimax Formulation	Simultaneous game: $\min_G \max_D V(D, G)$
Log-Likelihood Basis	Derived from binary classification using Bernoulli log-probabilities