

# Redes Generativas Antagónicas (GANs)

UNMSM – FISI – Unidad de Post Grado

Profesor: Juan Gamarra Moreno

# Introducción a GANs

# Concepto básico: ¿Qué son las GANs?

- Las Redes Generativas Antagónicas (Generative Adversarial Networks, GANs) son un tipo de modelo de aprendizaje profundo diseñado para generar datos sintéticos que imitan distribuciones de datos reales.
- **Propósito:** Crear datos nuevos, como imágenes, sonidos o texto, a partir de patrones aprendidos en un conjunto de datos existente.
- **Enfoque:** Se basa en la interacción entre dos redes neuronales que compiten entre sí: el **Generador** y el **Discriminador**.

# Arquitectura de las GANs

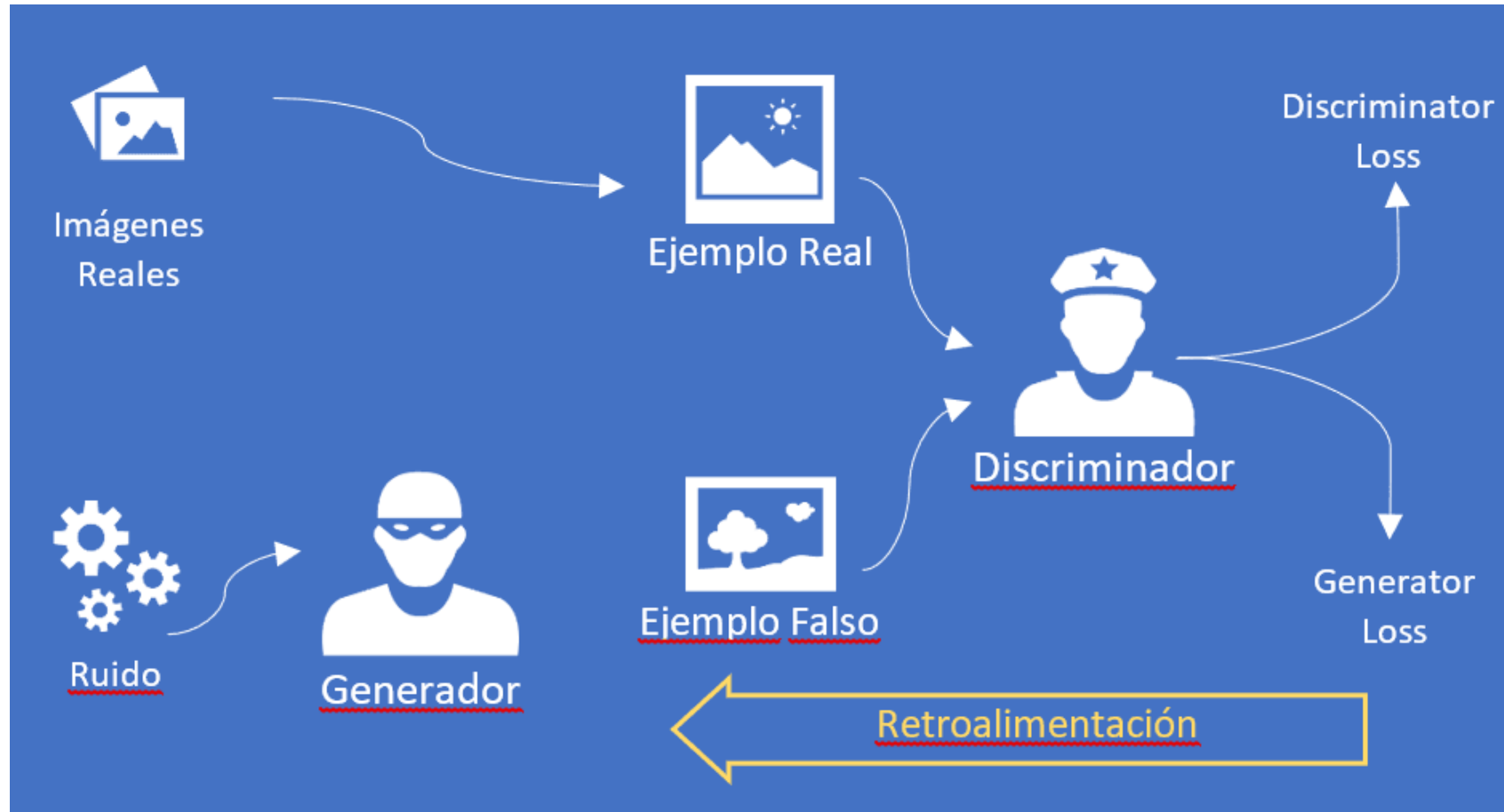
## 1. Generador (G):

- Es una red neuronal que toma como entrada un vector de ruido aleatorio (normalmente un vector de distribución gaussiana o uniforme).
- Su objetivo es transformar este ruido en datos sintéticos que sean similares a los datos reales.
- Ejemplo: Convertir ruido aleatorio en una imagen que parezca un rostro humano.

## 2. Discriminador (D):

- Es una red neuronal que toma como entrada tanto los datos reales como los datos generados.
- Su objetivo es clasificar si los datos son reales o generados por el generador.
- Actúa como un "crítico" que guía al generador para mejorar la calidad de sus datos sintéticos.

# Arquitectura de las GANs



# Proceso de Entrenamiento: El Juego Min-Max

El entrenamiento de GANs se basa en una dinámica adversaria entre las dos redes:

**1. Generador:** Aprende a engañar al discriminador produciendo datos sintéticos más realistas.

**2. Discriminador:** Mejora continuamente para diferenciar con mayor precisión entre datos reales y generados.

La función objetivo de una GAN es un juego de suma cero, donde el discriminador busca maximizar su precisión y el generador busca minimizar la capacidad del discriminador de distinguir datos reales de sintéticos. Matemáticamente:

$$\cdot \min_G \max_D \mathbb{E}_{x \sim P_{\text{real}}} [\log(D(x))] + \mathbb{E}_{z \sim P_{\text{ruido}}} [\log(1 - D(G(z)))]$$

# Desafíos Comunes en el Entrenamiento de GANs

## 1. Inestabilidad del Entrenamiento:

- Puede ser difícil para las GANs alcanzar un equilibrio entre el generador y el discriminador.
- A menudo, el generador o el discriminador domina al otro, lo que resulta en un modelo poco útil.

## 2. Colapso de Modo:

- El generador puede aprender a producir un rango limitado de resultados, ignorando otros modos en la distribución de datos reales.
- Esto lleva a datos sintéticos repetitivos y poco variados.

## 3. Convergencia Lenta:

- El proceso iterativo de ajuste entre el generador y el discriminador puede requerir muchos ciclos de entrenamiento para producir resultados satisfactorios.

## 4. Sensibilidad a Hiperparámetros:

- Las GANs son sensibles a la configuración de hiperparámetros como la tasa de aprendizaje y la arquitectura de las redes.

# Aplicaciones de GANs



# Generación de Imágenes

GANs son ampliamente utilizadas para crear imágenes realistas o artísticas:

- **Rostros:**

- Generación de rostros humanos que parecen reales, como en el sitio web [This Person Does Not Exist](#).

- **Paisajes:**

- Creación de paisajes ficticios para videojuegos o películas.

- **Arte:**

- Creación de obras de arte originales basadas en estilos preexistentes o generando nuevos estilos.

## Ejemplo:

GANs pueden entrenarse para generar fotografías de personas o paisajes inexistentes al aprender patrones en imágenes reales.

# Generación de Datos Sintéticos para Entrenamiento

- **Propósito:**

- Aumentar conjuntos de datos pequeños o equilibrar clases desbalanceadas en problemas de clasificación.

- **Ventaja:**

- Los datos generados por GANs pueden ser muy similares a los datos reales, lo que mejora el rendimiento de modelos de aprendizaje automático.

## **Ejemplo:**

En salud, GANs generan imágenes de resonancias magnéticas para entrenar modelos de diagnóstico sin necesidad de acceder a grandes cantidades de datos reales.

# Mejora de Resolución de Imágenes (Superresolución)

GANs, como las **SRGANs (Super-Resolution GANs)**, se utilizan para mejorar la calidad de imágenes de baja resolución:

- Transforman imágenes pixeladas en versiones más nítidas.
- Aplicaciones en restauración de fotografías antiguas o mejora de calidad en videos de baja resolución.

## **Ejemplo:**

Se pueden usar GANs para convertir imágenes de cámaras de baja calidad en fotografías claras y detalladas.

# Traducción de Estilo (De Bocetos a Imágenes Realistas)

GANs permiten transferir estilos entre dominios visuales distintos:

- Convertir bocetos simples en imágenes fotorrealistas.
- Transformar imágenes en blanco y negro en versiones coloreadas automáticamente.
- Aplicar estilos artísticos a fotografías (p. ej., convertir una foto al estilo de Van Gogh).

## **Ejemplo:**

Pix2Pix, una arquitectura basada en GANs, se usa para transformar bocetos arquitectónicos en modelos renderizados realistas.

# Creación de Contenido para Videojuegos, Diseño, etc.

- **Diseño de Niveles:**

- Generación automática de mapas, terrenos y elementos visuales para videojuegos.

- **Personajes y Objetos:**

- Creación de personajes y objetos tridimensionales que no requieren diseño manual.

## **Ejemplo:**

GANs son empleadas en juegos como Minecraft para generar terrenos dinámicos o en proyectos de diseño de contenido generado por inteligencia artificial.

# Laboratorio

# Configuración inicial y librerías necesarias

- Instalación y configuración de TensorFlow/Keras.
- Introducción rápida al manejo de datasets con MNIST o un conjunto de datos similar.

# Implementación básica de una GAN

- Creación del generador:
  - Una red neuronal simple que transforma un vector aleatorio en una imagen.
- Creación del discriminador:
  - Una red neuronal para clasificar imágenes reales y generadas.
- Combinar el generador y el discriminador para formar la GAN.



# Entrenamiento de la GAN

- Inicialización del bucle de entrenamiento:
  - Generación de ruido y producción de imágenes sintéticas.
  - Entrenamiento alternado entre generador y discriminador.
- Uso de pérdidas para guiar ambos modelos:
  - Pérdida del generador (confundir al discriminador).
  - Pérdida del discriminador (clasificar correctamente real vs. generado).

# Generación de imágenes sintéticas

- Al final de varias épocas de entrenamiento, visualizar imágenes generadas.
- Discutir los resultados y cómo mejorar la calidad.

# Actividades: Mejora del modelo

- Agregar capas adicionales al generador o discriminador para mayor complejidad.
- Ajustar hiperparámetros como la tasa de aprendizaje o el tamaño del ruido.