

Universidad Veracruzana

FACULTAD DE NEGOCIOS Y TECNOLOGÍAS

GENERATIVE ADVERSIAL NETWORK (GAN)

Servicio Social

Autor:

Fhatima Reyes Alejandre
Juan David Delgado Muñoz

Jueves 06 de Abril 2024

1 Introducción

Los modelos generativos han ganado terreno en el campo del aprendizaje automático, y en esta ocasión hablaremos sobre el modelo GAN (Generative Adversarial Network), creado por el investigador y científico "Ian Goodfellow" en el año 2014. El modelo GAN se consideró como aquel que le dio imaginación a las computadoras, dicho modelo se encuentra en el ramo de los modelos generativos no supervisados.

Muy por encima, podemos decir que el modelo GAN consta de dos redes neuronales profundas enfrentándose entre sí, así que veamos todo esto más a fondo.

1.1 Objetivos

Al finalizar la lectura se espera que puedas comprender lo siguiente:

1. Que es GAN.
2. Componentes que conforman a un modelo GAN.
3. Como funciona un modelo generativo GAN.
4. Tipos de GAN y sus aplicaciones.

2 Generative Adversarial Network (GAN)

2.1 ¿Qué es un modelo GAN?

Como se menciona de manera rápida en la introducción, un modelo GAN forma parte de los modelos generativos no supervisados, y tiene como función principal la creación de nueva información, para casos más prácticos: generar nuevas imágenes fake a partir de un conjunto de imágenes de entrenamiento, pero estas nuevas imágenes se generan a tal grado de dificultar demasiado la decisión de si dichas nuevas imágenes son verdaderas o falsas, pero, ¿Cómo se logra esto? El modelo GAN consta de dos redes neuronales profundas combatiendo entre sí.

2.2 Componentes que conforman a un modelo GAN

Como se observa en la Figura 1, el modelo GAN está formado principalmente por dos redes neuronales profundas conocidas como "Generador" y "Discriminador", así como otros componentes que serán descritos a continuación.

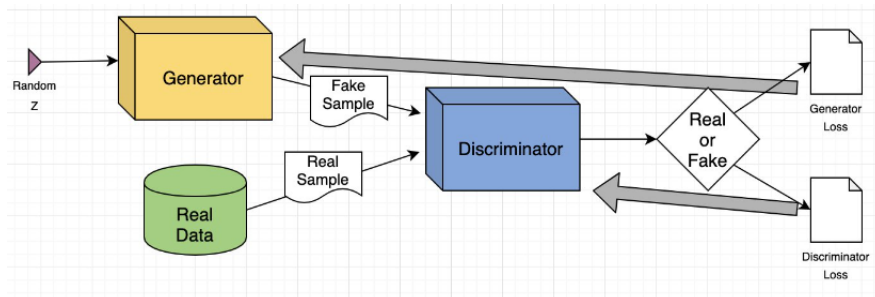


Figure 1: Estructura del modelo GAN (AWS, 2023).

- Random z: También conocido como ruido, es generalmente una distribución aleatoria normal.
- Generador: Tiene como objetivo crear una imagen apartir de la distribución normal dada como entrada, de una forma tan real como sea posible, con la intención de engañar al discriminador, con la única condición de que el generador no deberá tener acceso al dataset de entrenamiento.
- Dataset de entrenamiento (Real Data): Es el conjunto de datos (imagenes en este caso) reales utilizadas para el entreamiento del discriminador.
- Discriminador: Es un clasificador con solo dos salidas a partir de la imagen de entrada, mostrando si dicha imagen es real o falsa.

3 ¿Cómo funciona un modelo generativo GAN?

Primero se debe generar n cantidad de ejemplos para "z" (los cuales indicaran el tamaño de un batch), estos valores serán introducidos en el generador dandonos como resultado n cantidad de imagenes fakes.

Después se deberá obtener otro batch con la misma cantidad de n fotos del dataset de entrenamiento. Una vez obtenidos estos dos grupos con la misma cantidad de imagenes (imagenes fakes e imagenes reales) cada uno, se procede a realizar el entrenamiento del discriminador (se calcula el error y se realiza la retropropagación con el descenso del gradiente).

Una vez terminado el entrenamiento del discriminador, volvemos a generar n cantidad de ejemplos para z, estos valores se introducen nuevamente en el generador obteniendo n cantidad de imagenes fake, pero en esta ocasión se deben etiquetar como si fueran verdaderas. Estas nuevas imagenes seran ingresadas

en el discriminador para que este nos diga si realmente son falsas o verdaderas, posteriormente se calcula nuevamente el error y se realiza la retropropagación con el descenso del gradiente, pero esta vez en el generador.

Todos estos pasos se repiten según el número de epoch establecidos.

Se considera que el modelo alcanza su estado óptimo cuando este no pueda clasificar si la imagen de entrada en el discriminador es real o falsa (50 por ciento de probabilidad para opcion).

4 Tipos de GAN y sus aplicaciones

Los modelos GAN se pueden clasificar según las fórmulas matemáticas utilizadas y la forma de interacción entre el generador y el discriminador. Algunas de las más conocidas son:

- Vanilla GAN.
- Conditional GAN.
- Deep Convolutional GAN.
- Super-Resolution GAN.

Algunas de sus aplicaciones son:

- Generación de rostros.
- Farmacología.
- Ciberseguridad.

5 Explicación detallada del funcionamiento interno de un modelo GAN

5.1 Estructura

El Generador será el encargado de construir imagenes fakes de la siguiente manera:

- Recibe como entrada dos números random generados en el rango de una distribución normal.

- los valores de entrada son propagados hacia adelante en cada capa de la red hasta llegar a la salida, obteniendo 784 valores, los cuales serán transformados a una matriz de (28, 28), esta matriz será la imagen fake obtenida.
- Nuestro generador se compone de dos capas ocultas, de las cuales, a la primera se le aplica la función de activación ReLU y a la segunda la función Sigmoide.

El discriminador o clasificador será el encargado de decirnos que tanta probabilidad existe de que una imagen de entrada sea verdadera o falsa. En nuestro caso se realiza lo siguiente:

- El discriminador recibe como entrada un array de 784 posiciones, teniendo en cuenta que es nuestra imagen de 28x28 píxeles, pero aplanados en una sola dimensión.
- los valores de entrada son propagados hacia adelante en cada capa de la red hasta llegar a la salida, obteniendo 1 único valor comprendido de 0 a 1, el cual nos indicará la probabilidad de que la imagen de entrada sea verdadera o falsa.
- Nuestro discriminador se compone de dos capas ocultas, de las cuales, a la primera se le aplica la función de activación ReLU y a la segunda la función Sigmoide.

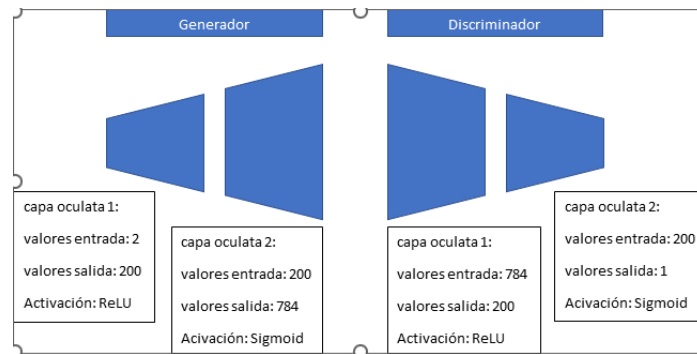


Figure 2: Estructura utilizada para el ejemplo

Es importante mencionar que el modelo explicado presenta características que nos dieron buenos resultados, por lo que los parámetros y funciones pueden variar según las necesidades.

5.2 Entrenamiento

En este ejemplo trabajamos con imagenes en escala de grises, por lo que el valor asignado a cada pixel se encontrará de 0 a 255. Basandonos en documentación, encontramos que el proceso de entrenamiento se realiza mediante mini lotes de imagenes, una vez dicho esto veamoslo más a detalle.

- Primero se establece un numero de epoch o vueltas que dara nuestro modelo.
- Se necesita calcular la cantidad de pasos que se daran en cada epoch, esto se realiza dividiendo la cantidad de imagenes en el dataset de entrenamiento entre el tamaño del minibatch, por ejemplo, tengo 8 imagenes de entrenamiento y quiero que mi modelo aprenda de 2 en 2, por lo que este deberá de dar 3 pasos, de [0:2], [2:4], [4:8] en cada epoch.
- Primero se deberá entrenar al discriminado, por lo que en cada paso se realiza lo siguiente:
 - 1.-se generan n muestras normales aleatorias (tomando en cuenta que n es la longitud que tiene cada lote de imagenes, en nuestro caso 2)
 - 2.-las n muestras obtenidas se ingresan al generador para obtener n imagenes fake etiquetadas como falsas (0).

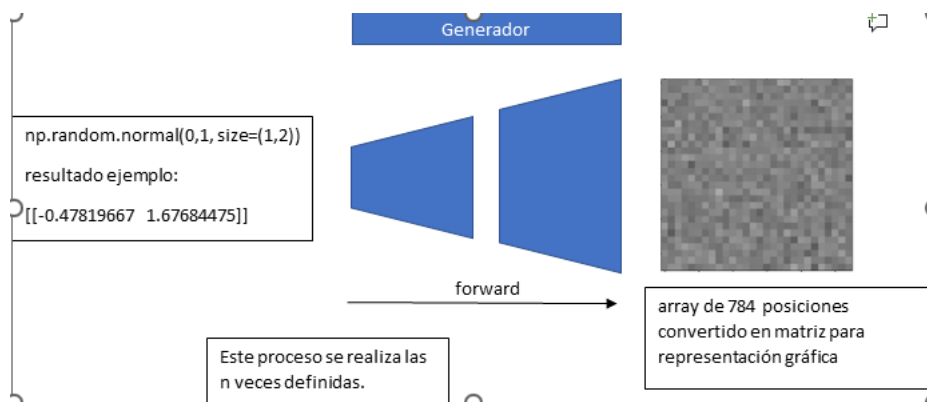


Figure 3: Generación de n imagenes fake

3.-Del dataset de imagenes reales se toma la muestra con la misma n cantidad de imagenes.En este caso tendríamos 2 imagenes reales y 2 falsas con las que sen entrenara el discriminador, se calcula el error restando el resultado de la predicción menos el valor esperado, el error es propagado

hacia atras y se ajustan los pesos, para esto se hace uso del optimizador Adam.

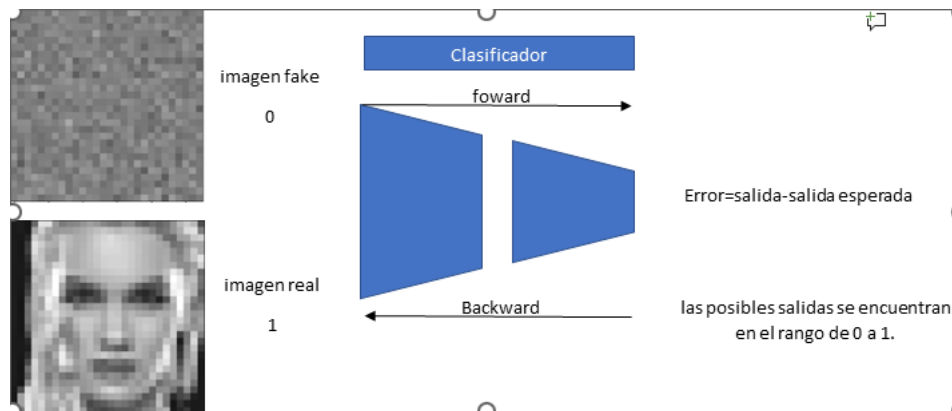


Figure 4: Entrenamiento del discriminador

4.-Posteriormente se procede a entrenar al generador, para esto primero se generan 2 muestras normales aleatorias, estas muestras son introducidas en el generador para generar 2 imagenes fake, solo que en esta ocasion serán etiquetadas como verdaderas.

5.-Las imagenes generadas son introducidas en el discriminador para ver como son clasificadas, se calcula el error restando el resultado menos el valor esperado, recordando que en estos intentos se espera que las imagenes proporcionadas por el generador se vuelvan tan buenas que el discriminador no sepa si son verdaderas o falsas.

6.-El error calculado es propagado hacia atras por todas las capas del discriminador, pero esta vez no se actualizan sus pesos 7.-Del resultado del back propagation en el discriminador, obtenemos los gradientes necesarios para realizar la propagación hacia atras en la red generadora y poder actualizar sus pesos y sesgos.

8.-todo esto se repite durante todos los pasos calculados hasta completar el epoch

- Todos los puntos mencionados se repiten hasta completar el número de epochs establecidos.

Notas importantes:

El optimizador del generador se establecio con un learning rate de 0.001, mientras que el optimizador para el discriminador se establecio con un valor de

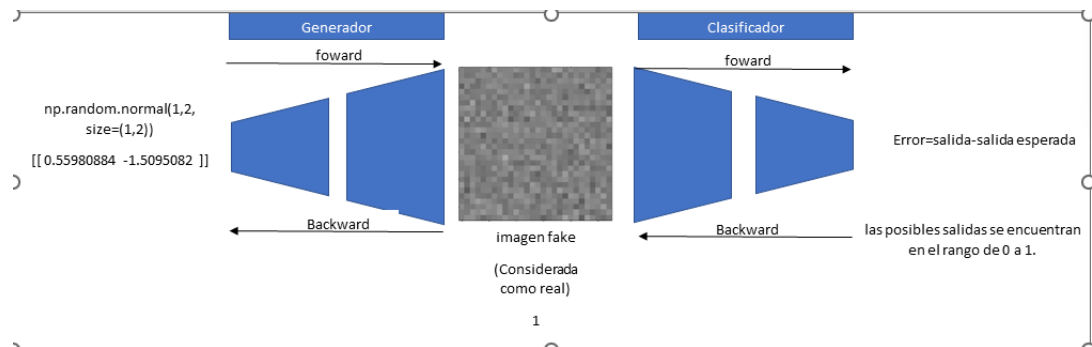


Figure 5: Entrenamiento del Generador (En este punto los pesos y sesgos del discriminador no se actualizan)

0.0002.

El total de epochs o ciclos establecidos para el modelo completo fue de 1000.

Las imagenes utilizadas fueron de 28x28 pixeles, en escala de grises.

Con estas características obtuvimos los siguientes resultados:

Dataset de números MNIST:



Figure 6: Imagenes generadas por el modelo GAN entrenado con el dataset de MNIST

Dataset de estrellas:

Dataset de ropa FASHION:

Pequeño conjunto de rostros:

Si bien con los rostros se obtienen imagenes demasiado pixeleadas, recordemos que la entrada del generador es de solo dos valores, y existen metodos como la Super Resolución para obtener mejores resultados.

Para mejorar estas imagenes de entreno una red neuronal que nos permite aumentar el pixelaje de una imagen pequeña (28x28 pixeles) a una mayor resolución (160x160 pixeles). Por lo que al pasar las imagenes obtenidas con el modelo GAN, en nuestra red aumentadora de resolución generamos lo siguiente:

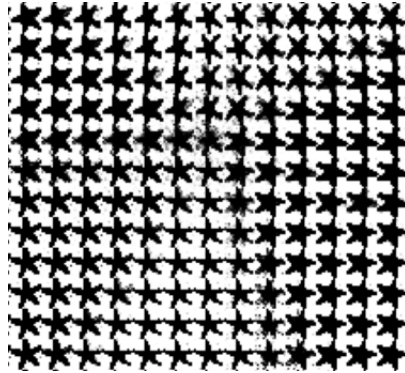


Figure 7: Imagenes generadas por el modelo GAN entrenado con un dataset de estrellas

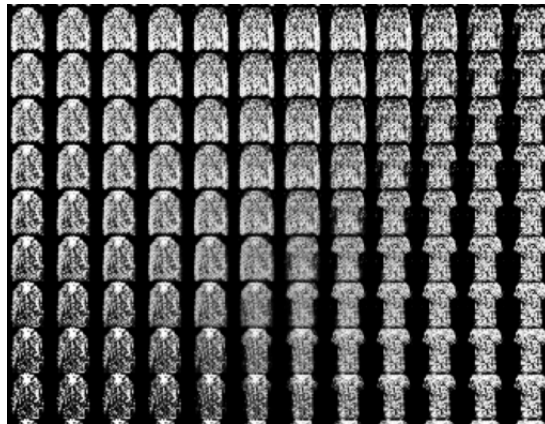


Figure 8: Imagenes generadas por el modelo GAN entrenado con el dataset FASHION

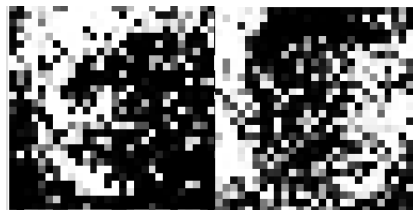


Figure 9: Imagenes generadas por el modelo GAN entrenado con ocho rostros distintos



Figure 10: Mejora de resolución de las imágenes generadas por el modelo GAN

6 Bibliografías

- Velázquez, C., V. (2022, 15 junio). Redes Generativas Antagónicas: Una pequeña explicación y aplicaciones. Medium. <https://medium.com/mcd-unison/redes-generativas-antagónicas-una-pequeña-explicación-y-aplicaciones-6727bdda0a39>
- ¿Qué es una GAN? - Explicación sobre las redes generativas antagónicas - AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/gan/>
- Redes Adversarias (Generative Adversarial Networks): explicación y tutorial en Python — Codificando Bits. (s. f.). Codificando Bits. <https://www.codificandobits.com/blog/redes-adversarias-explicacion-y-tutorial-python/>