

## **Texto en maquina a apariencia escrito a mano usando cGAN y TensorFlow**

El presente proyecto tiene como objetivo entrenar un modelo de red generativa adversarial condicional (cGAN, por sus siglas en inglés) que sea capaz de generar imágenes que imiten la escritura manuscrita de letras del alfabeto. Utiliza como fuente de datos el conjunto de datos EMNIST (Extended MNIST), en particular la partición "letters", que contiene letras manuscritas de A-Z, en mayúsculas y minúsculas. Parte del código usado tanto de inspiración como de ayuda es de la siguiente pagina <https://www.kaggle.com/code/hamiddamadi/conditional-gan-on-mnist>.

La meta última es lograr que el modelo genere letras manuscritas realistas condicionadas a una entrada textual. De esta manera, dado un texto como "HOLA", el sistema debe poder generar una imagen en la que aparezca esa palabra escrita a mano.

### **Carga y preparación del conjunto de datos**

El dataset EMNIST/letters es cargado usando la librería `tensorflow_datasets`, que permite acceder fácilmente a conjuntos de datos estandarizados. La carga se hace en dos divisiones: entrenamiento y prueba por facilidad, pero realmente solo se usa la de entrenamiento. Las imágenes vienen etiquetadas del 1 al 26, donde cada número representa una letra del alfabeto ya sea en mayúscula o minúscula.

Estas imágenes son luego convertidas a arreglos de NumPy para facilitar su manipulación y visualización. Asimismo, se normalizan para que sus valores estén en el rango  $[0, 1]$ , lo cual es una práctica común para mejorar el aprendizaje de redes neuronales.

### **Diseño del modelo cGAN**

Una cGAN consta de dos componentes:

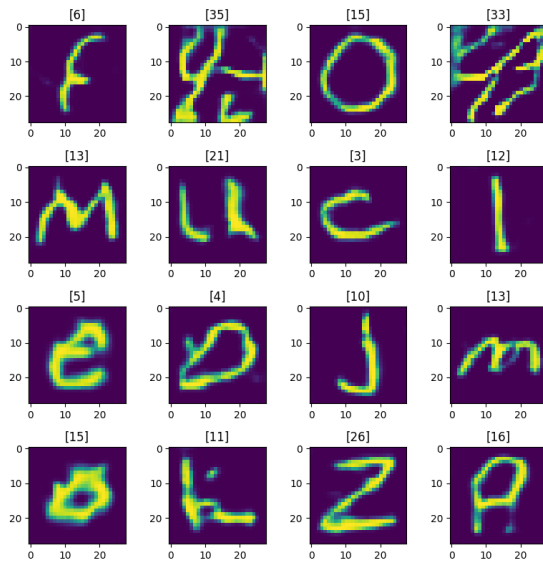
1. Generador: toma como entrada un vector aleatorio (ruido) y una etiqueta de clase (letra) y genera una imagen correspondiente a esa clase.
2. Discriminador: recibe como entrada una imagen y una etiqueta, y predice si la imagen es real (del conjunto de datos) o falsa (generada).

Ambos modelos son entrenados en conjunto en una especie de "juego" donde el generador trata de engañar al discriminador, y el discriminador intenta detectar las falsificaciones.

La dimensión latente (latent\_dim) representa el tamaño del vector de ruido aleatorio que alimenta al generador. Este vector permite introducir variabilidad en las imágenes generadas para una misma letra. Por ejemplo, con el mismo label "A" y diferentes vectores latentes, el generador puede producir varias versiones manuscritas distintas de la letra "A".

## **Entrenamiento del modelo**

Durante el entrenamiento, se generan pares de datos:



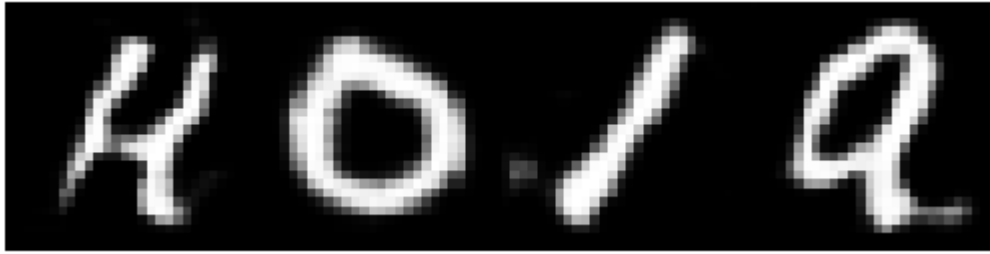
- Imagen real con su etiqueta

- Imagen falsa generada con su etiqueta correspondiente

El discriminador aprende a distinguir entre ambas, y el generador aprende a crear imágenes que el discriminador no pueda diferenciar de las reales. Y cada cierta época, se genera un conjunto de letras para observar visualmente el progreso del modelo. EL modelo se entreno por alrededor de 20,000 épocas y esto fue la ultima imagen de parte de lo que aprendió.

## **Generación de texto manuscrito**

Una vez entrenado el modelo, se puede usar el generador para producir imágenes de letras manuscritas individuales. Para formar palabras o frases completas, simplemente se genera cada letra por separado y luego se unen horizontalmente en una sola imagen. Esto se logra generando un vector de ruido por letra, ingresando su etiqueta correspondiente (por ejemplo, la letra "A" corresponde al label 1) y luego usando el generador para crear la imagen. Finalmente, todas las letras generadas se combinan visualmente para producir el resultado deseado: una imagen que parece contener un texto manuscrito, aunque en realidad fue completamente generado por una red neuronal. Aquí un ejemplo del resultado final.



### **Por ultimo**

Este proyecto demuestra cómo las redes generativas adversariales condicionales pueden ser usadas para tareas de generación de imágenes sintéticas a partir de texto. Al usar EMNIST como fuente de datos, se entrena un generador capaz de imitar la escritura humana para cada letra del alfabeto. Con este modelo, es posible construir herramientas que transformen texto digital en escritura "a mano", lo cual podría tener aplicaciones creativas, educativas o de accesibilidad.