



Generative adversarial network

Materia:

Inteligencia artificial avanzada para la ciencia de datos II (Gpo 50)

Integrantes:

José Antonio Miranda Baños

Campus: Querétaro.

Fecha: 1 Nov de 2024

Generative adversarial network for D&D characters

José Antonio Miranda Baños

IRS, Instituto Tecnológico y de Estudios
Superiores de Monterrey
(Campus QRT)
Qrt, Mx
A01611795@tec.mx

Abstract—

Keywords— ML, IA, Convolución, Generativo, Pytorch

I. INTRODUCCIÓN

El presente reporte tiene como objetivo principal examinar y documentar el proceso de creación de un avatar de personaje aleatorio para el juego de rol Dungeons & Dragons (D&D) mediante el uso de una red Generative Adversarial Network (GAN). Este proyecto aborda un desafío significativo, ya que la creación de avatares en D&D implica una gran variedad de posibilidades y combinaciones de características, reflejo de la rica diversidad en las razas, clases, personalidades y aspectos visuales que definen a los personajes en este juego. La complejidad de representar esa diversidad a través de un modelo GAN hace que este proyecto sea tanto ambicioso como técnicamente desafiante. Para este reporte nos basaremos en la implementación que muestra Pytorch [1]

Las GAN, conocidas por su capacidad para generar imágenes sintéticas a partir de distribuciones de datos, no son modelos triviales de entrenar. Requieren un equilibrio delicado entre el generador y el discriminador, dos redes que compiten entre sí para mejorar en cada iteración. Entrenar una GAN con éxito demanda un enfoque cuidadoso para ajustar hiperparámetros, gestionar la calidad de las imágenes y evitar problemas comunes como el colapso de modo, donde el modelo puede estancarse y generar imágenes repetitivas. En este sentido, la meta de este reporte no solo consiste en crear un avatar específico, sino en explorar las capacidades de la GAN para representar fielmente la diversidad de personajes en D&D, evaluando el rendimiento y las limitaciones de la red en este contexto.

A lo largo de este trabajo, documentaré los diferentes experimentos realizados, los ajustes en el modelo y los resultados obtenidos, reflejando los avances y obstáculos encontrados. Cabe mencionar que, aunque hasta el momento no he logrado obtener un avatar completamente satisfactorio para usar como mi propio personaje en la campaña semanal de los sábados, los resultados alcanzados aportan perspectivas valiosas para mejorar el modelo. Este proceso iterativo busca no solo perfeccionar el avatar final, sino también contribuir al entendimiento de cómo las GAN pueden utilizarse en aplicaciones creativas complejas y orientadas a un público específico, como es el caso de los jugadores de D&D.

II. ¿QUE ES UNA GAN?

Una GAN (Generative Adversarial Network) es un framework de Deep Learning que se enfoca en aprender una distribución de datos para luego generar nuevos ejemplos que

se ajusten a esa distribución, de modo que los resultados producidos puedan ser indistinguibles de los datos reales. Este tipo de modelo es ampliamente utilizado en la generación de imágenes sintéticas, creación de arte digital, mejora de resolución de imágenes, y otras aplicaciones en las que es importante que los datos generados se asemejen a muestras reales. La arquitectura de una GAN está compuesta por dos modelos que interactúan entre sí: el **Generador** y el **Discriminador**, cuya dinámica competitiva es esencial para el proceso de aprendizaje.

A) Generador

El Generador es la parte de la GAN responsable de crear nuevos datos, en este caso, imágenes que simulan las del conjunto de entrenamiento. Su objetivo es generar muestras que imiten de manera realista las características visuales de los datos originales, de tal forma que resulten creíbles para el Discriminador. Al principio del entrenamiento, las imágenes generadas suelen ser de muy baja calidad y fácilmente reconocibles como "falsas" por el Discriminador. Sin embargo, a medida que el entrenamiento avanza y el Generador recibe retroalimentación, sus imágenes van mejorando y se vuelven progresivamente más parecidas a las reales.

B) Discriminador

El Discriminador, por su parte, actúa como un "juez" que clasifica las imágenes que recibe como "reales" o "falsas". Su tarea es determinar si una imagen pertenece o no al conjunto de entrenamiento original. En términos prácticos, el Discriminador evalúa cada imagen generada por el Generador y asigna una probabilidad de que sea auténtica. Durante el entrenamiento, el Discriminador también se entrena con imágenes reales para mejorar su habilidad de distinguir entre los datos del conjunto de entrenamiento y las falsificaciones generadas por el Generador.

Para este punto se intuye la dirección que toma este modelo. Esta dinámica competitiva entre el Generador y el Discriminador es lo que define la esencia de una GAN. A medida que el Generador intenta crear imágenes suficientemente convincentes para "engañar" al Discriminador, este último se vuelve cada vez más efectivo en identificar imágenes falsas. Este ciclo de mejora mutua se convierte en un juego en el que el Generador y el Discriminador compiten hasta alcanzar un punto de equilibrio. En teoría, cuando ambos modelos han alcanzado un estado de aprendizaje óptimo, las imágenes generadas son tan realistas que el Discriminador tiene dificultades para distinguir las de las reales.

Sin embargo, debido a esta interdependencia entre los dos modelos, entrenar una GAN puede ser muy complejo y delicado. Existen varios desafíos típicos, como el **colapso de modo**, que ocurre cuando el Generador produce solo un pequeño subconjunto de imágenes similares y no representa toda la diversidad de los datos de entrenamiento. También puede suceder que uno de los modelos, ya sea el Generador o el Discriminador, se vuelva demasiado efectivo, lo cual desequilibra el sistema y frustra el entrenamiento. En algunos casos, el modelo no logra converger y ambos submodelos oscilan sin llegar a un equilibrio, produciendo resultados inconsistentes. Estos problemas comunes hacen que el entrenamiento de una GAN requiera una cuidadosa optimización y monitoreo constante para evitar que las imágenes generadas se vuelvan aleatorias o irreales.

III. DATA SET

Ahora que sabemos cómo funcionan las GAN, el primer paso para construir una, sería encontrar un conjunto de datos de preferencia no aleatorios para entrenar nuestra GAN, mientras más mejor. En nuestro caso ocuparemos un data set del proyecto “Dungeons and Diffusion”. [2]



Este data set viene con alrededor de 2.5k imágenes de 512x512 de diversas especies del juego. Nuestro objetivo será poder crear un personaje que pueda encajar dentro de este set de entrenamiento por lo que podemos hacer una lista de las cosas que esperamos que tenga este personaje basado en los rasgos comunes entre todas las imágenes.

- 2 extremidades
- 1 cabeza
- Color diferente al fondo
- 1 o varias armas

Si por el contrario te gustaría generar un personaje mas específico te recomendamos generar GAN con sets de entrenamiento más cercanos a lo que estas buscando o revisar la implementación de ‘Justin’ en ‘Dungeons and Diffusion’.

IV. IMPLEMENTACIÓN

Como primera implementación usaremos la siguiente configuración para nuestros modelos.

A. Generador

REFERENCIAS

- [1] DCGAN Tutorial — PyTorch Tutorials 2.5.0+cu124 documentation. (n.d.). https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html
- [2] *OxJustin/Dungeons-and-Diffusion · Datasets at Hugging face*. (n.d.). <https://huggingface.co/datasets/OxJustin/Dungeons-and-Diffusion>
- [3]