

Red neuronal convolucional desde cero para clasificar personajes de One Piece

Integrantes.

| A. Badilla Olivas | B80874 |

| Brandon Alonso Mora Umaña | C15179 |

| Joseph Stuart Valverde Kong | C18100 |

El código en este repositorio esta inspirado en el contenido en las fuentes referenciadas abajo. Muchos de los ejemplos encontrados funcionan solo para imágenes binarias en blanco y negro. Este repositorio contiene código que espera clasificar imágenes del anime One Piece en sus respectivos personajes. El código está escrito en Python y utiliza la biblioteca cupy para ejecutarse en la GPU. El código aún está en desarrollo y no está listo para su uso.

Esencialmente para resolver el problema de la identificación se construyo una red convolucional con varias capas que luego pasa por una red normal de varias capas, generando. La red neuronal guarda su estado después de una serie de entrenamiento.

El conjunto de datos utilizado es el [clasificador de imágenes de One Piece](#) de Kaggle. El conjunto de datos cuenta con 650 imágenes por personaje, las imágenes fueron redimensionadas a 256x256 píxeles. 18 personajes.

Resultados.

Se logró hacer que tanto el forward pass y backwards pass funcionaran para cada tipo de capa, esto quiere decir que, despues de procesar una imagen, este propaga hacia atrás el error utilizando descenso de gradiente.

Pudimos notar que la red reducía el error con cada época. Tuvimos el problema que al procesar las 11000 imágenes que componían, pues tomaba mucho tiempo. Logramos hacer que el error llegara 2.35 con 25 épocas totales, entrenandola por 2 días.

Como se preprocesaron las imágenes.

Image_processor es lo que utilizamos para transformar los datos.

Entrenamiento.

El archivo one_piece_classifier.py es el que se encarga de entrenar la red neuronal. Pide por linea de comandos el path a la carpeta con las imágenes ya procesadas (imagenes de one piece de 256x256) y pregunta si se quiere entrenar la red neuronal desde cero o si se quiere cargar un modelo previamente entrenado.

Utilizar el modelo solo para prededcir.

El archivo onepiece_classifier.py es el que se encarga de utilizar el modelo ya entrenado para predecir. Pide por linea de comandos el path a la carpeta con las imágenes ya procesadas (imagenes de one piece de 256x256).

Referencias:

- 3Blue1Brown. (2017a, 5 de octubre). [Pero, ¿qué es una red neuronal? | Capítulo 1, Aprendizaje profundo](#) [Video]. YouTube.
- 3Blue1Brown. (2017b, 16 de octubre). [Descenso del gradiente, cómo aprenden las redes neuronales | Capítulo 2, Aprendizaje profundo](#) [Video]. YouTube.
- 3Blue1Brown. (2017c, 3 de noviembre). [Cálculo de retropropagación | Capítulo 4, Aprendizaje profundo](#) [Video]. YouTube.
- 3Blue1Brown. (2017d, 3 de noviembre). [¿Qué está haciendo realmente la retropropagación? | Capítulo 3, Aprendizaje profundo](#) [Video]. YouTube.
- Nielsen, M. A. (2015). [Redes neuronales y aprendizaje profundo](#).
- Clasificador de imágenes de One Piece. (28 de mayo de 2022). [Kaggle](#).
- Russell, S., & Norvig, P. (2021). Inteligencia Artificial: Un Enfoque Moderno, Edición Global. Pearson Higher Ed.
- Universidad de Standford. (2023). [CS231N Redes Neuronales Convolucionales para Reconocimiento Visual](#).

- StatQuest con Josh Starmer. (2021a, 8 de febrero). [Redes neuronales Parte 5: ArgMax y SoftMax](#) [Video]. YouTube.
- StatQuest con Josh Starmer. (2021b, 8 de febrero). [La derivada de SoftMax, paso a paso!!!](#) [Video]. YouTube.
- StatQuest con Josh Starmer. (2021c, 1 de marzo). [Redes Neuronales Parte 6: Entropía Cruzada](#) [Video]. YouTube.
- StatQuest con Josh Starmer. (2021d, 1 de marzo). [Redes Neuronales Parte 7: Derivadas de Entropía Cruzada y Retropropagación](#) [Video]. YouTube.
- StatQuest con Josh Starmer. (2021e, 8 de marzo). [Redes Neuronales Parte 8: Clasificación de Imágenes con Redes Neuronales Convolucionales \(CNNs\)](#) [Video]. YouTube.
- The Independent Code. (2021, 23 de mayo). [Red Neuronal Convolucional desde Cero | Matemáticas & Código Python](#) [Video]. YouTube.
- Weidman, S. (2019). Aprendizaje Profundo desde Cero: Construyendo con Python desde los Primeros Principios. O'Reilly Media.