INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

Inteligencia artificial avanzada para la ciencia de datos II (Gpo 502)

Modulo 2: Deep Learning

“Entregable: Implementación de un modelo de deep learning ”

Jose Pablo Cobos Austria **A01274631**

Profesor:
Dr. Benjamín Valdés Aguirre

Santiago de Querétaro, 25 de noviembre del 2022

Introducción

Descripción del problema

Antes de empezar a generar nuestros modelo de redes neuronales, es necesario comprender correctamente cual es el problema o situación al cual tratamos de dar solución con nuestra propuesta.

“La navidad es una temporada donde la mayoría de los niños reciben regalos, siendo la mayoría juguetes, que van desde muñecas, figuras de acción, bicicletas, etc, y entre todos ellos encontramos los LEGO, que son bloques con los que los niños pueden armar cualquier que se imaginen. Por esa razón lo que se quiere lograr es que se implemente un modelo de reconocimiento de imágenes usando deep learning para identificar qué piezas tienen.“

Desarrollo

Informacion del Dataset

El dataset con el que se va a trabajar se obtuvo de la página de Kaggle, teniendo originalmente un total de más de 200 piezas de LEGO diferentes, con un total de 4000 imágenes, no obstante por practicidad y utilidad, se recortó el dataset a simplemente 6 piezas de LEGO y cada una con 1500 imagenes para entrenar y 100 de prueba.

Además, para el desarrollo de la actividad se hizo uso de librerías de Tensorflow con Keras para generar las capas de nuestro modelo de red neuronal.

Código

Lo primero que hacemos importar las librerías que utilizamos en el desarrollo de nuestro modelo, con un gran énfasis en la librería de numpy, cv2 y tensorflow, ya que estas tres son la que mayor importancia tienen para la actividad ya que nos ayudarán a analizar las imágenes, realizarles ajustes, pasar a arreglos, manipular estos mismos y finalmente para poder generar nuestro modelo de red neuronal

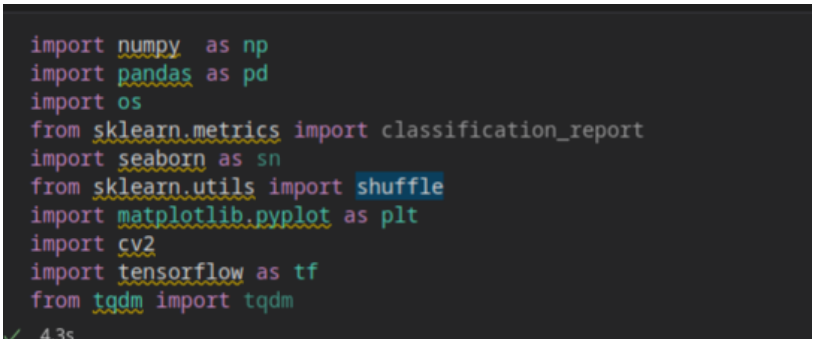

Imagen1. *Importar librerías*

Después de eso, lo que realizamos fue primero crear un arreglo con los nombres de cada uno de los bloques que habíamos seleccionado como categoría, después de eso fueron enumerados y finalmente imprimimos cada categoría con su debida etiqueta.

Para la carga de los datos, lo que se realizó fue una función, donde lo primero que hacía era cargar la dirección donde se encontraba la carpeta de los datos de entrenamiento y los datos de prueba, y después en un ciclo lo que se iba haciendo era entrar a cada uno de las carpetas anteriormente mencionadas e ir entrando a las subcarpetas de las categorías que tenían las imágenes, y estas mismas eran leías y se formateaban, se les cambiaba de color y el tamaño con el fin de qué todos tuvieran el mismo estándar, finalmente esto se agregaban a un arreglo donde estaba la imagen en formato de arreglo de imagenes e etiquetas . Al terminar la

función lo que te regresaba era un arreglo que tenía tanto las imágenes en forma de arreglo, como sus etiquetas.

Ahora para la parte de análisis de nuestros datos, lo primero que hacemos con las imágenes y las etiquetas entrenamiento lo que hacemos es mezclarlas, haciendo un shuffle. Y posteriormente imprimimos cuantas muestras había de entrenamiento y cuánto sería de prueba además de qué tamaño tenía cada una de las imágenes.

Aparte de eso gráfica vamos de forma breve un poco cómo es que está la distribución de nuestros samples, y podemos observar que en efecto las imágenes de entrenamiento son de 1500 y las imágenes de prueba son solamente 100 por cada una de las categorías.

Finalizado eso, lo siguiente que se realizó fue otra función donde se imprimieron algunas de las muestras que teníamos para nuestro entrenamiento con su etiqueta para poder observar que de verdad todo estaba en orden antes de entrenar nuestro modelo.

Modelo 1

En la sección de generación de modelo, el primer modelo que se propuso fue un modelo donde se tenían unas cinco capas: la primera de ellas era de Conv2D para dividir en pequeñas capas de 2D, luego tenemos una capa de Maxpooling que realiza la discretización de la muestra , pasamos una capa flaaten donde aplanaban las capas a 1D, y finalizamos con dos capas Dense, una de ellas con una función de activación relu y otra soft as para conectar las capas.

Analiza los resultados de tu modelo set de pruebas y validación.

Mejora tu modelo usando técnicas de regularización, ajustando hiper parámetros, modificando la arquitectura de tu modelo o buscando otro modelo.

Documenta y explica cuáles son los cambios que funcionaron y por qué funcionaron.

Prueba tu implementación con un set de datos y realiza algunas predicciones. Las predicciones las puedes correr en consola o las puedes implementar con una interfaz gráfica apoyándote en los visto en otros módulos.

Conclusiones

Referencias

- s/f. (s/a). *B200C LEGO Classify Conv2D.* Kaggle. Recuperado el 25/11/22 de: