

Deep Convolutional Generative Adversarial Network (DC-GAN) generadora de imágenes

Proyecto Final

Dominguez Juarez Jorge Uriel, 427186, ju.dominguezjuarez@ugto.mx

Martínez Miranda Emilio, 346316, e.martinezmiranda@ugto.mx

Introducción

En este proyecto se implementó una DC-GAN mediante el uso de Keras|Tensorflow. La red neuronal tiene como objetivo la generación de imágenes después de ser entrenada con un dataset de una categoría de imágenes.

Las Redes Neuronales constituyen una rama de Machine Learning y, dentro de esta rama, se encuentran distintas categorías tal como las Generative Adversarial Networks (GANs). Esta arquitectura tiene una amplia gama de tareas, dentro de las cuales se tiene la generación de imágenes. Para esta tarea en específico se crearon las Deep Convolutional GANs (DC-GAN), las cuales utilizan capas convolucionales y mejoran la resolución de las imágenes en comparación a las GANs convencionales.

Objetivo

Mostrar el funcionamiento de una red neuronal con la arquitectura DCGAN en la generación de imágenes basadas en el dataset de entrenamiento.

Justificación

La importancia de este proyecto recae sobre la aplicación del conocimiento adquirido durante el curso sobre las redes neuronales, específicamente la arquitectura de una red neuronal convolucional, y la investigación realizada sobre la arquitectura de las redes generativas adversariales. La conjunción de ambas arquitecturas permite el desarrollo de la DCGAN implementada, por lo que se obtienen mejores resultados en comparación del uso de una sola arquitectura.

Por otro lado, este proyecto tiene como fundamento la exploración de diferentes aplicaciones de Machine Learning, siendo la generación de datos una de ellas. Por lo que al implementar métodos y arquitecturas poco vistas durante el curso se refuerza y amplía significativamente el aprendizaje adquirido.

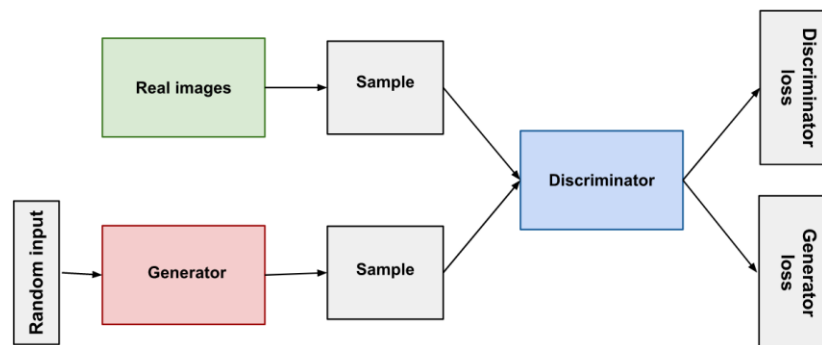
Marco Teórico

Las redes adversarias generativas (GAN) son una clase de técnicas de aprendizaje automático que consisten en dos modelos entrenados simultáneamente: uno (el Generador) entrenado para generar datos falsos, y el otro (el Discriminador) entrenado para discernir los datos falsos de los reales ejemplos. (Langr & Bok, 2019, p. 5)

Una red generativa adversaria (GAN) tiene dos partes:

- **El generador** aprende a generar datos posibles. Las instancias generadas se convierten en ejemplos de entrenamiento negativos para el discriminante.
- **El discriminador** aprende a distinguir los datos falsos del generador de los reales. El discriminante penaliza al generador por producir resultados improbables.

Cuando comienza el entrenamiento, el generador produce datos obviamente falsos y el discriminante aprende con rapidez que es falso. A medida que el entrenamiento avanza, el generador se acerca al resultado que puede engañar al discriminador. Por último, si el entrenamiento del generador funciona bien, el discriminante empeora al indicar la diferencia entre real y falso. Comienza a clasificar los datos falsos como reales y disminuye su exactitud.



El generador y el discriminante son redes neuronales. El resultado del generador se conecta directamente a la entrada del discriminador. Mediante la propagación inversa, la clasificación del discriminador proporciona una señal que el generador usa para actualizar sus pesos.

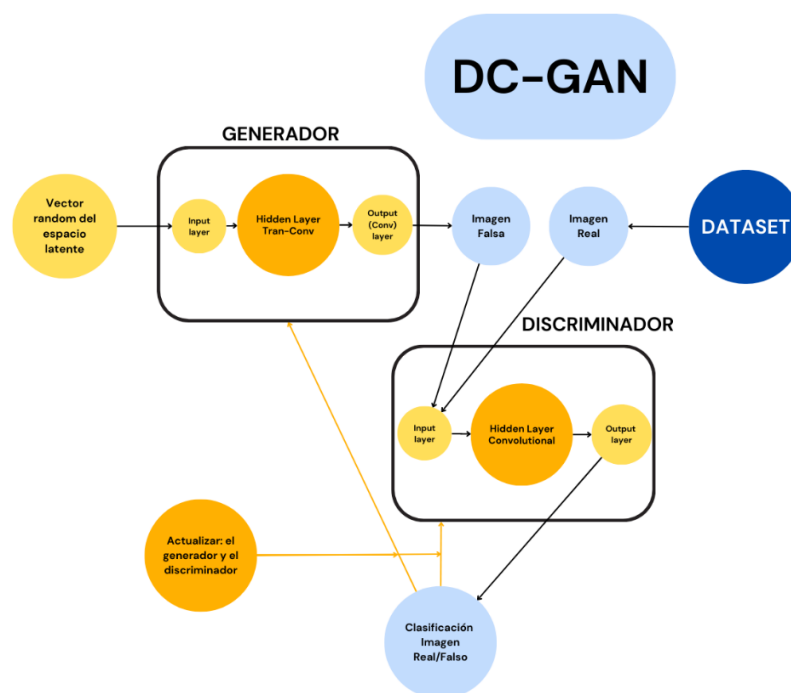
Después podemos hablar de una rama que se deriva de la principal que son las GANs, las cuales son las DCGAN es decir el termino DC viene de Deep Convolutional lo cual se refiere a que la DCGAN utiliza también convolucionales es decir en el generador y discriminador de nuestra GAN hay convolucionales ayudando a la obtención de mejores resultados, entonces definamos lo que hace las convolucionales, lo que realiza una convolucional es hacer un “barrido” de la imagen con filtros o también llamados kernels, los cuales recorren toda la imagen en sus diferentes canales ya sea las propiedades de las imágenes y así genera una

nueva representación de la imagen filtra con ciertos kernels, una sola capa convolucional puede contener cientos de filtros, es el motivo principal de las redes convolucionales es aprender estos filtros cada vez hacerlos mejor para poder obtener características de las principales de las imágenes como lo son bordes , texturas, fondo, todo esto gracias a algo que se conoce como correlación espacial que se tienen en las imágenes.

Esto dentro de la DCGAN es muy importante ya que su principal uso es la generación de nuevas imágenes dentro de un dataset y las redes convolucionales son perfectas para trabajar con imágenes, por lo tanto, la combinación de GAN y Convolucionales dio como resultados muy buenos en el área de generación de imágenes, pero entonces como funciona la DCGAN pues lo principal es el cambio entre el generador y discriminador de una GAN convencional.

Ahora el generador de nuestra red tendrá dentro una red convolucional transpuesta que lo que hace será recibir un vector y mediante la profundidad de la red convolucional convertirla en imagen que será el uno de los inputs a nuestro discriminador(junto a las imágenes del dataset) y su objetivo será hacer cada vez imágenes que se parezcan al dataset y así persuadir a él discriminador, ahora el discriminador de nuestra red tendrá dentro una red convolucional común que lo que hará será de una imagen dada de un tamaño definido la tomará y a partir de la profundidad de las convoluciones la clasificará en imagen falsa o real y a partir de eso actualizaran el generador y el discriminador para realizar mejor sus tareas . (Langr & Bok, 2019)

Diagrama de bloques



Se utilizaron distintos datasets para el entrenamiento de la DCGAN. Dos de ellos fueron los datasets Caltech 101 (Li, Andreeto, Ranzato, & Perona, 2022) y Caltech 256 (Griffin, Holub, & Perona, 2022), obtenidos desde Caltech Data by Caltech Library; los cuales contienen distintas categorías de imágenes. Otros datasets, la mayoría de los utilizados, fueron extraídos de la plataforma Kaggle.

Independientemente del dataset, todas las imágenes se preprocesaron para reescalarlas y que todas tuvieran un tamaño de 64x64. Posteriormente los datos de las imágenes de baja resolución se vectorizaron en un arreglo y se normalizaron entre 255 debido a los valores RGB (rango que va de 0 a 255). Además, se reescalaron los datos de entrada, o datos de las imágenes, para que pasaran de un rango de 0 a 1 hacia un rango de -1 a 1 para utilizar de manera adecuada la función de activación tanh.

Se utilizó una arquitectura Deep Convolutional GAN, la cual utiliza un Generador y un Discriminador con varias capas convolucionales. Esto se realizó basándose en la guía de desarrollador oficial de Keras, (Chollet, 2020), y en un artículo acerca del uso de GANs para generar datos en la página de Towards Data Science, (Dobilas, 2022).

De manera específica, el generador se constituye de un modelo secuencial de 4 capas ocultas (una capa Densa y 3 de convolución transpuesta con función de activación ReLU) las cuales generan imágenes a partir de los datos de entrada (vectores de datos de las imágenes de entrenamiento) y una capa de salida (con una operación de convolución y función de activación tanh) que genera que la imagen pase a un formato de colores RGB.

Por otro lado, el discriminador está constituido por un modelo secuencial de 3 capas ocultas (todas de convolución con una función de activación Leaky ReLU) que reciben las imágenes y las procesan; una capa de Flatten, Dropout y de salida que vectorizan los datos de las imágenes. Con esto se realiza una clasificación de los datos para determinar si las imágenes son reales o falsas, es decir, pertenecen al dataset o son producidas por el generador.

Para definir la función de entrenamiento de la DCGAN se generaron distintas funciones. La primera tiene objetivo de recuperar distintas imágenes del dataset original con una etiqueta de clase 1. La segunda tiene como objetivo generar puntos en el espacio latente para que la tercera función produzca imágenes con una etiqueta de clase 0, a partir de los puntos en el espacio latente.

La definición de la función de entrenamiento consiste en separar un lote de muestras de entrenamiento en 2, mitad y mitad. Una mitad consiste en imágenes reales y otra de imágenes falsas, con ambas mitades el discriminador actúa y realiza la clasificación de los datos de entrada. Al mismo tiempo el generador produce muestras falsas que deben pasar por reales dentro del discriminador, por lo que el generador mejora el desempeño según la clasificación del discriminador.

Finalmente se entrena la DCGAN con un dataset y se muestra el desempeño cada 100 épocas. Estos entrenamientos pueden almacenarse en archivos .h5 para poder ser cargados posteriormente. Y, para observar el resultado, se muestran diferentes imágenes generadas por el método implementado.

Actividades de programación

Actividad	Integrante
Obtención de datasets desde Caltech	EMM
Obtención de datasets desde Kaggle	JUDJ
Importación de librerías	JUDJ
Carga y preprocesamiento de datasets	EMM
Definición del generador	EMM
Definición del discriminador	JUDJ
Definición de la GAN	EMM
Definición de las funciones de muestreo	JUDJ
Definición de función de entrenamiento de la GAN	EMM Y JUDJ
Entrenamiento de la GAN con distintos datasets	EMM y JUDJ
Evaluación del desempeño con distintos datasets	EMM y JUDJ

Referencias

Referencias

- Chollet, F. (4 de December de 2020). *The Sequential model*. Obtenido de https://keras.io/guides/sequential_model/
- Dobilas, S. (27 de Junio de 2022). *GANs: Generative Adversarial Networks - An Advanced Solution for Data Generation*. Obtenido de <https://towardsdatascience.com/gans-generative-adversarial-networks-an-advanced-solution-for-data-generation-2ac9756a8a99>
- Griffin, G., Holub, A., & Perona, P. (6 de Abril de 2022). *Caltech 256*. Obtenido de <https://doi.org/10.22002/D1.20087>
- Langr, J., & Bok, V. (2019). *Gans in Action: Deep Learning with Generative Adversarial Networks*. New York: Manning.
- Li, F.-F., Andreeto, M., Ranzato, M., & Perona, P. (6 de Abril de 2022). *Caltech 101 (1.0)*. Obtenido de <https://doi.org/10.22002/D1.20086>