

Tarea N°4:

Aprendizaje Profundo Avanzado

Cristóbal Tomás David Alberto Lagos Valtierra

Escuela de Ingeniería, Universidad de O'Higgins

02 de Noviembre del 2023

Abstract—Este informe presenta tres notebooks que exploran diversas aplicaciones de la inteligencia artificial utilizando las bibliotecas TensorFlow y Keras. El primer notebook aborda el uso de autoencoders para comprimir y reconstruir imágenes, destacando su capacidad para reducir dimensiones y eliminar ruido. En el segundo notebook, se explora la potencia de las Redes Adversarias Generativas (GANs) para generar imágenes realistas, centrado en el reconocimiento de dígitos manuscritos. Por último, el tercer notebook introduce la transferencia neuronal de estilo, una técnica que combina contenido y estilo para crear salidas artísticas, utilizando un modelo VGG19 preentrenado. Estos notebooks ofrecen una visión integral del aprendizaje profundo, desde la compresión de imágenes hasta la generación y transformación creativa de contenido visual.

I. INTRODUCCIÓN

La inteligencia artificial ha emergido como una disciplina fascinante y poderosa, impulsada por una diversidad de enfoques y técnicas que buscan replicar la capacidad humana de aprender, razonar y generar. En este informe, exploraremos tres áreas clave de la IA mediante la implementación de algoritmos y modelos en el entorno de TensorFlow y Keras.

En primer lugar, abordaremos el mundo de los autoencoders, una técnica de aprendizaje no supervisado que ha demostrado ser fundamental en la comprensión y representación eficiente de datos. A través de un análisis detallado, observaremos cómo los autoencoders pueden comprimir y reconstruir imágenes, destacando su papel en la reducción de dimensionalidad y la eliminación de ruido visual.

Posteriormente, nos adentraremos en las Redes Adversarias Generativas (GANs), una innovadora técnica que emplea la competencia entre dos modelos, un generador y un discriminador, para generar datos realistas. A través de la implementación práctica, exploraremos cómo las GANs pueden aprender a generar imágenes que se asemejan a conjuntos de datos específicos, enfocándonos en el contexto del reconocimiento de dígitos manuscritos.

Finalmente, nos adentraremos en la transferencia neuronal de estilo, una técnica que fusiona la visión artística con la inteligencia artificial. Desarrollaremos una comprensión detallada de cómo esta técnica utiliza modelos preentrenados para extraer representaciones de estilo y contenido, permitiendo la creación de imágenes que combinan características visuales únicas.

A través de estos tres enfoques, este informe busca proporcionar una visión integral de las capacidades y aplicaciones de la inteligencia artificial, desde la compresión y generación de datos hasta la fusión creativa de estilos visuales. Cada

sección ofrece una experiencia práctica, respaldada por la implementación de modelos en el marco de trabajo de TensorFlow y Keras, proporcionando una plataforma sólida para la comprensión y exploración activa de estas emocionantes disciplinas.

II. MARCO TEÓRICO

A. Machine Learning

Machine Learning (ML) es una rama de la inteligencia artificial (IA) que se centra en el desarrollo de algoritmos y modelos que permiten a las máquinas aprender patrones y realizar tareas sin ser explícitamente programadas. En lugar de seguir instrucciones específicas, los sistemas de ML utilizan datos para aprender y mejorar su rendimiento con el tiempo.

En Machine Learning, podemos identificar ciertos componentes que son imprescindibles en esta:

- 1) **Datos:** Por norma general se dividen en conjuntos de entrenamiento, prueba y validación. La calidad y cantidad de datos afectan directamente el rendimiento del modelo.
- 2) **Modelo:** Un modelo de ML es la representación matemática de un sistema o proceso. Puede ser una red neuronal, un árbol de decisiones, entre otros.
- 3) **Algoritmos:** Los algoritmos son procedimientos matemáticos que guían el entrenamiento del modelo. Estos ajustan los parámetros del modelo en función de los datos de entrada.
- 4) **Entrenamiento:** Durante el entrenamiento, el modelo se ajusta iterativamente a los datos de entrenamiento para minimizar la diferencia entre las predicciones y las etiquetas reales.

Estos son componentes clave en el uso de cualquier técnica dentro de Machine Learning, básicamente ningún modelo que omita uno o más componentes de estos puede funcionar correctamente.

B. Autoencoders

Los autoencoders son modelos de aprendizaje automático pertenecientes a la familia de las redes neuronales que se utilizan para la reducción de dimensionalidad y la reconstrucción de datos. Estos modelos están compuestos por dos partes principales: un codificador, que reduce la entrada a una representación más compacta llamada "dimensión latente", y un decodificador, que reconstruye la entrada original a partir de esta representación.

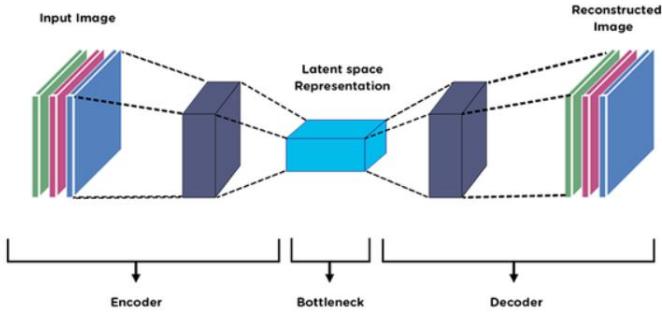


Fig. 1: Estructura de un Autoencoder.

Podemos verla como un proceso compuesto por 2 fases (codificación y decodificación), y la representación comprimida entremedia (dimensión latente). En la fase de codificación, el autoencoder reduce la dimensión de la entrada mediante capas de reducción, como capas completamente conectadas o convolucionales, generando así la representación latente. La dimensión latente es una representación comprimida y abstracta de la entrada original. Contiene características clave que permiten la reconstrucción de la entrada. La fase de decodificación utiliza capas que aumentan la dimensión para reconstruir la entrada original. El objetivo es que la reconstrucción sea lo más fiel posible a la entrada original.

Los autoencoders tienen una amplia gama de aplicaciones, a continuación se detallan algunas:

- 1) **Compresión de Datos:** los autoencoders se pueden utilizar en aplicaciones de compresión de datos, donde aprenden representaciones eficientes para comprimir y luego reconstruir la información original.
- 2) **Eliminación de Ruido:** pueden ser empleados para eliminar ruido de datos. Un autoencoder entrenado para reconstruir datos ruidosos aprende a preservar solo la información más relevante.
- 3) **Generación de Contenido:** se pueden aplicar en la generación de contenido, como la creación de imágenes realistas a partir de una representación latente.
- 4) **Transferencia de Estilo:** los autoencoders pueden aprender a codificar y decodificar el estilo y contenido de imágenes, permitiendo la creación de nuevas obras de arte.



Fig. 2: Ejemplo Transferencia de estilo.

C. Redes Adversarias Generativas

Las Redes Adversarias Generativas (GANs) son un tipo de arquitectura de red neuronal que se utiliza para la generación de datos nuevos y realistas. Este enfoque innovador se basa en la competencia entre dos modelos, un generador y un discriminador, que se entrenan simultáneamente mediante un proceso adversarial.

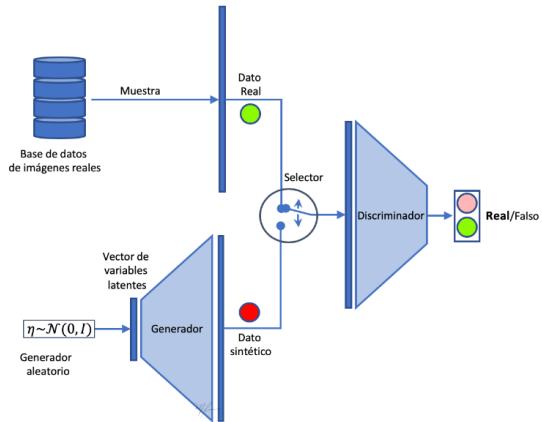


Fig. 3: Estructura Redes Adversarias Generativas.

Podemos verla como una competencia entre 2 modelos (Generador y Discriminador) en un determinado proceso (Proceso Adversarial). El generador crea datos, como imágenes, a partir de una distribución de ruido aleatorio. Su objetivo es generar datos que sean indistinguibles de los datos reales. El discriminador evalúa la autenticidad de los datos, determinando si provienen del conjunto de datos real o fueron generados por el generador. Su tarea es mejorar su capacidad para distinguir entre lo real y lo generado. Durante el entrenamiento, el generador busca mejorar constantemente su habilidad para engañar al discriminador, mientras que este último mejora su capacidad para discernir entre lo real y lo generado.

Las GANs tienen una amplia gama de aplicaciones, a continuación se mencionan algunas:

- 1) **Generación de Imágenes:** son conocidas por generar imágenes fotorrealistas, como retratos humanos, paisajes o incluso obras de arte.
- 2) **Mejora de Resolución:** pueden utilizarse para mejorar la resolución de imágenes, generando versiones más detalladas a partir de versiones de baja resolución.
- 3) **Generación de Caras Realistas:** pueden utilizarse para generar retratos humanos fotorrealistas, incluso de personas que no existen. Esta aplicación es valiosa en la industria de videojuegos y simulación.
- 4) **Superresolución de Imágenes Médicas:** GANs pueden mejorar la resolución de imágenes médicas, como tomografías computarizadas o resonancias magnéticas, proporcionando imágenes más detalladas y precisas.

III. METODOLOGÍA

A. Autoencoders

En primera instancia, se define un autoencoder básico que solamente tendrá como función codificar y decodificar las entradas que se le entreguen. Se cargan los datos, se aplica una normalización y se dividen en conjunto de entrenamiento y de prueba. Se define un autoencoder con dos capas densas: un encoder, que comprime las imágenes en un vector latente de 64 dimensiones, y un decoder, que reconstruye la imagen original a partir del espacio latente. A continuación, se entrena el modelo utilizando el conjunto de entrenamiento como entrada y como objetivo. El encoder aprenderá a comprimir el conjunto de datos de 784 dimensiones al espacio latente, y el decoder aprenderá a reconstruir las imágenes originales. En el encoder se usa una función de activación relu y en el encoder se usa sigmoid, se entrena el modelo usando como optimizador a adam. Ahora que el modelo está entrenado, se pone a prueba codificando y decodificando imágenes del conjunto de prueba.

Como segundo ejemplo, se entrenará un autoencoder para eliminar el ruido de las imágenes en el conjunto de datos Fashion MNIST, demostrando así una aplicación práctica de los usos que se le pueden aplicar a los autoencoders. Se entrenará el autoencoder utilizando la imagen ruidosa como entrada y la imagen original como objetivo. Para lograr este cometido, se implementará un autoencoder convolucional usando capas Conv2D en el encoder, y capas Conv2DTranspose en el decoder y se entrenará este modelo utilizando el optimizador adam y se ejecutarán un total de 10 épocas.

En el encoder, la capa de entrada especifica que las imágenes de entrada tendrán dimensiones (28, 28, 1), lo que indica imágenes en escala de grises (1 canal) de tamaño 28x28 píxeles. Además, tiene dos capas convolucionales 2D con activación ReLU. La primera capa tiene 16 filtros, un tamaño de kernel de (3, 3), activación ReLU, relleno ('same') y un paso (strides) de 2. La segunda capa tiene 8 filtros con configuración similar.

En el decoder, están presentes dos capas de convolución transpuesta (upsampling). La primera tiene 8 filtros, un tamaño de kernel de (3, 3), un paso de 2, activación ReLU y relleno ('same'). La segunda capa tiene 16 filtros con configuración similar. Además, tiene una capa de convolución 2D con un solo filtro, un tamaño de kernel de (3, 3), activación sigmoide y relleno ('same'). Esta capa produce la salida reconstruida de la imagen.

B. Redes Adversarias Generativas

Primeramente, se Cargan y preparan el conjunto de datos a utilizar, en este caso se utilizará el conjunto de datos MNIST para entrenar el generador y el discriminador. El generador generará dígitos manuscritos parecidos a los datos MNIST. Se crean los modelos utilizando la Keras Sequential API. El generador utiliza capas tf.keras.layers.Conv2DTranspose (upsampling) para producir una imagen a partir de una semilla (ruido aleatorio). Comienza con una capadense que toma esta semilla como entrada, luego realiza un upsampling varias veces hasta alcanzar el tamaño de imagen deseado de 28x28x1. El discriminador es un clasificador de imágenes basado en

Redes neuronales convolucionales (CNN). Se utiliza el discriminador (aún no entrenado) para clasificar las imágenes generadas como reales o falsas. El modelo se entrenará para dar valores positivos a las imágenes reales y negativos a las falsas.

Se definen funciones de pérdida y optimizadores para ambos modelos. Se utiliza un método de pérdida del discriminador para cuantificar la capacidad del discriminador para distinguir las imágenes reales de las falsas. Compara las predicciones del discriminador sobre imágenes reales con una matriz de 1s, y las predicciones del discriminador sobre imágenes falsas (generadas) con una matriz de 0s. Por otro lado, la pérdida del generador cuantifica su capacidad para engañar al discriminador. Intuitivamente, si el generador funciona bien, el discriminador clasificará las imágenes falsas como reales (o 1). Aquí se comparan las decisiones del discriminador sobre las imágenes generadas con una matriz de 1s.

En adición, se agregan funciones como la de agregar checkpoints, que sirven para guardar y restaurar modelos, lo que puede ser útil en caso de que se interrumpe una tarea de entrenamiento de larga duración. También implementa los bucles de entrenamiento que comienza con un generador que recibe una semilla aleatoria como entrada. Esta semilla se utiliza para producir una imagen. A continuación, se utiliza el discriminador para clasificar imágenes reales (extraídas del conjunto de entrenamiento) e imágenes falsas (producidas por el generador). La pérdida se calcula para cada uno de estos modelos, y los gradientes se utilizan para actualizar el generador y el discriminador.

Finalmente, se generan y guardan las imágenes. Se llama al método train() definido previamente para entrenar el generador y el discriminador simultáneamente. Entrenar las GANs puede ser complicado, ya que es importante que el generador y el discriminador no se dominen mutuamente (por ejemplo, que se entrenen a un ritmo similar).

Al principio del entrenamiento, las imágenes generadas parecen ruido aleatorio. A medida que avanza el entrenamiento, los dígitos generados parecen cada vez más reales. Después de unas 50 épocas, se parecen a los dígitos MNIST. Cada época toma una cantidad considerable de tiempo según la configuración del entorno en el que se ejecute. Por último, se crea un GIF utilizando imageio utilizando las imágenes guardadas durante el entrenamiento.

C. Transferencia neuronal de estilos

Se busca implementar la técnica de optimización Transferencia neuronal de estilos, que hace uso de una imagen de contenido y otra de referencia de estilo y se mezclan para que la imagen de salida se parezca a la imagen de contenido, pero "pintada" con el estilo de la imagen de referencia de estilo. Esto se consigue optimizando la imagen de salida para que coincida con las estadísticas de contenido de la imagen de contenido y las estadísticas de estilo de la imagen de referencia de estilo. Estas estadísticas se extraen de las imágenes mediante una red convolucional.

En primer lugar, se importan e implementan los módulos que serán utilizados. Se parte Definiendo las representaciones

de contenido y estilo, para esto se utilizan las capas intermedias del modelo para obtener las representaciones de contenido y estilo de la imagen. Empezando por la capa de entrada de la red, las activaciones de las primeras capas representan características de bajo nivel como bordes y texturas. A medida que se avanza por la red, las últimas capas representan características de nivel superior, partes de objetos como ruedas u ojos. En este caso, está utilizando la arquitectura de red VGG19, una red de clasificación de imágenes preentrenada. Estas capas intermedias son necesarias para definir la representación del contenido y el estilo a partir de las imágenes. Para una imagen de entrada, se debe hacer coincidir las representaciones objetivas de estilo y contenido correspondientes en estas capas intermedias. Se eligen capas intermedias de la red para representar el estilo y el contenido de la imagen, se escogen estas, puesto que de esta forma pueden comprender la imagen al construir una representación interna que convierta los píxeles de la imagen en bruto en una comprensión compleja de las características presentes en la imagen.

Teniendo todo lo anterior, se puede construir el modelo, partiendo por especificar las entradas y salidas. Una vez creado el modelo se pueden aplicar técnicas extracción de estilos y contenidos, resulta que el estilo de una imagen puede ser descrito por las medias y correlaciones entre los diferentes mapas de características. Se realiza calculando una matriz de Gram incluyendo esta información tomando el producto exterior del vector de características por sí mismo en cada posición y promediando ese producto exterior en todas las posiciones. Esta matriz de Gram se puede calcular para una capa en particular como:

$$G_{LCD} = \sum_{ij} F_{lij}^c(x) F_{lij}^d(x) IJ$$

Teniendo estos fundamentos, podemos extraer estilo y contenido de imágenes y así, armar la construcción de un modelo que devuelva los tensores de estilo y contenido, así. Cuando se invoca sobre una imagen, este modelo devuelve la matriz grama (estilo) de las style_layers y el contenido de las content_layers. Teniendo este extractor de estilo y contenido, ahora podemos implementar el algoritmo de transferencia de estilo. Para ello, se calcula el error cuadrático medio de la salida de su imagen con respecto a cada objetivo y, a continuación, se toma la suma ponderada de estas pérdidas, de esta forma estamos realizando un descenso de gradiente. Una desventaja de esta implementación básica es que produce muchos artefactos de alta frecuencia. Para afrontar esto se utiliza un término de regularización explícito en los componentes de alta frecuencia de la imagen. En la transferencia de estilo, esto se denomina a menudo pérdida de variación total. De esta forma, tenemos todos los fundamentos que utiliza tensorflow al utilizar la Transferencia Neuronal de estilos.

IV. RESULTADOS

A. Autoencoders

Teniendo modelada la estructura de los autoencoders, y haciendo el respectivo entrenamiento, podemos mostrar ejemplos de las imágenes de entrada y salida en estos:

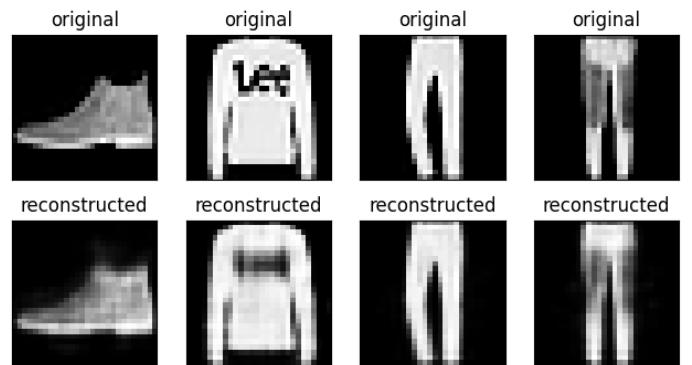


Fig. 4: Funcionamiento de un Autoencoder.

Lo mismo para el autoencoder convolucional que elimina el ruido de las imágenes:

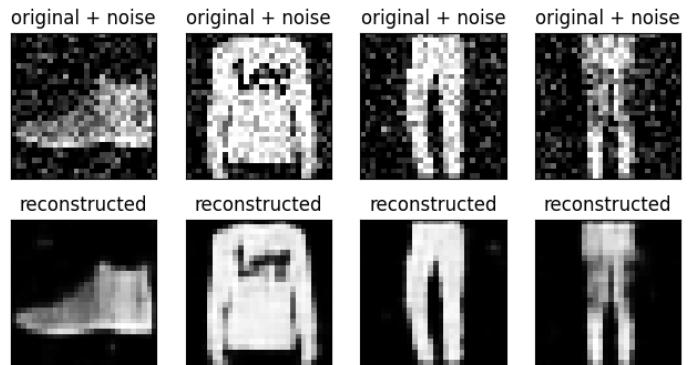


Fig. 5: Funcionamiento de un Autoencoder convolucional.

B. Redes Adversarias generativas

Teniendo modelada la estructura e implementada las distintas técnicas que utilizarán nuestras GAN como lo es el guardado de checkpoints o bucles de entrenamiento, podemos poner en funcionamiento la "lucha" entre el generado y discriminador para generar las imágenes:

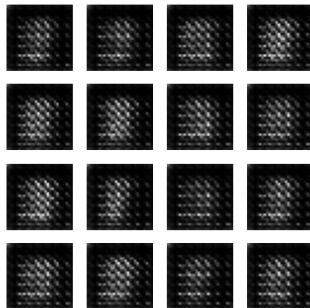


Fig. 6: Primera época de la GAN.

Tal como se aprecian en las imágenes, a medida que aumentan las épocas cada vez más se van pareciendo a los números, podemos iterar el número de épocas hasta que el discriminador no sea capaz de determinar si se trata de una imagen falsa o real.

C. Transferencia de Estilos

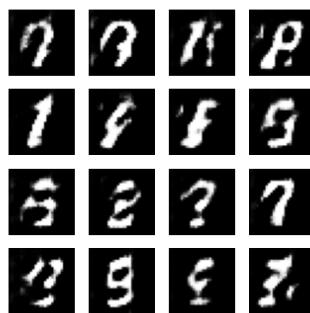


Fig. 7: Décima época de la GAN.

Teniendo modelada la estructura e implementada las distintas técnicas que utilizarán nuestra optimización como lo es la captura de contenidos y estilo o descenso de gradiente, podemos poner en funcionamiento nuestro optimizador para realizar una transferencia de estilos:



Fig. 10: Imagen con contenido.

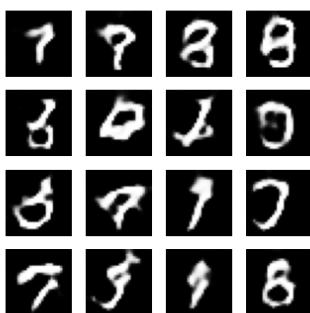


Fig. 8: Trigésima época de la GAN.



Fig. 11: Composición 7 de Wassily Kandinsky que servirá como estilo.

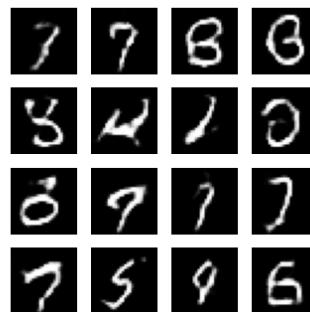


Fig. 9: Quincuagésima época de la GAN.

Teniendo la imagen, el contenido y las implementaciones para realizar las optimizaciones, podemos hacer uso de estas técnicas y aplicarlas:



Fig. 12: Transferencia de estilos con una optimización de corta duración.

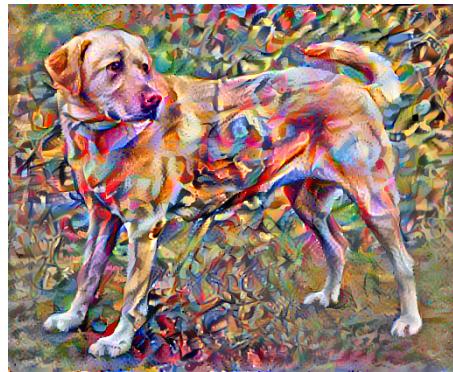


Fig. 15: Transferencia de estilos TensorFlow.



Fig. 13: Transferencia de estilos con una optimización de larga duración.

Una desventaja de la implementación realizada es que produce muchos artefactos de alta frecuencia.

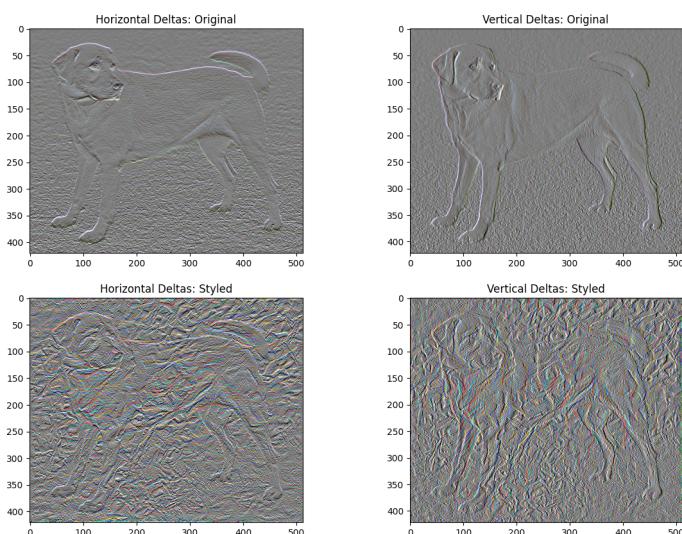


Fig. 14: Artefactos de alta frecuencia.

Esto muestra cómo han aumentado los componentes de alta frecuencia. Además, este componente de alta frecuencia es básicamente un detector de bordes. Se puede obtener un resultado similar del detector de bordes Sobel

V. ANÁLISIS Y DISCUSIONES

La construcción de autoencoders resultó muy interesante, destacando la libertad que brindan al permitir la definición de su estructura, funciones de activación y formato de entrada y salida. Esta flexibilidad es crucial, ya que diferentes problemas pueden requerir enfoques específicos. La capacidad de personalizar la arquitectura del encoder y decoder proporciona una herramienta poderosa para comprimir datos, recuperar los datos y realizar determinadas aplicaciones.

La relación entre las GANs y redes neuronales convolucionales (CNNs) sin dudad permite una gran variedad en la generación de imágenes realistas. La intrínseca interacción entre el generador y el discriminador permite entrenar modelos capaces de producir imágenes que son cada vez más difíciles de distinguir de los reales, principio que se usa bastamente en lo conocido como videos deepfake.

La implementación de la técnica de transferencia de estilos representa un puente entre el contenido y la estética visual. Al utilizar representaciones intermedias de capas en una red preentrenada, se logra capturar tanto el contenido como el estilo de imágenes específicas. Este enfoque no solo muestra la versatilidad de las técnicas de optimización, sino que también ofrece una herramienta poderosa para la creación de obras visuales únicas.

En conjunto, estos enfoques enriquecen el panorama de la inteligencia artificial. Estas técnicas ilustran la amplitud de aplicaciones y el impacto que la inteligencia artificial puede tener en la creación y manipulación de contenido visual. Su aplicación no solo destaca avances tecnológicos, sino también la convergencia entre ciencia y creatividad en el campo de la informática.

VI. CONCLUSIONES GENERALES

En este informe, exploramos tres grandes aplicaciones del aprendizaje profundo, utilizando herramientas avanzadas como TensorFlow y Keras. Las conclusiones generales destacan el impacto y la versatilidad de estas técnicas en el ámbito del aprendizaje profundo y la generación de contenido visual.

La implementación de autoencoders revela su potencial para comprimir información, eliminar ruido y generar representaciones significativas en el espacio latente. La capacidad de personalización de su estructura brinda flexibilidad, permitiendo adaptarse a diversas problemáticas. La relación entre GANs

y redes neuronales convolucionales destaca la capacidad de generar imágenes cada vez más realistas. Estas aplicaciones encuentran utilidad en la creación de datos sintéticos de alta calidad, con implicaciones en campos como la generación de imágenes y la síntesis de datos. La técnica de transferencia de estilos fusionando contenido y estética visual demuestra la capacidad de las redes neuronales preentrenadas para capturar y combinar características de diferentes imágenes. Este enfoque añade una dimensión artística, mostrando cómo la inteligencia artificial puede contribuir a la creación visual.

En conjunto, estos enfoques ilustran la diversidad de aplicaciones que el aprendizaje profundo ofrece. Desde la compresión y generación de imágenes con autoencoders hasta la creación de contenido visual realista mediante GANs, y la fusión artística de estilos con la transferencia neuronal.

Este informe resalta la intersección entre la ciencia de datos y la creatividad. La capacidad de estas técnicas para transformar datos en información valiosa y producir arte digital demuestra el impacto positivo y multifacético que la inteligencia artificial puede tener en diversas disciplinas. Este viaje es solo una ventana a las posibilidades infinitas que la inteligencia artificial promete en el futuro.

VII. BIBLIOGRAFÍA

- 1) http://personal.cimat.mx:8181/~mrivera/cursos/aprendizaje_profundo/dcgan/dcgan.html
- 2) <https://www.wolfram.com/language/12/machine-learning-for-images/built-in-image-style-transfer.html.es?footer=lang#:~:text=La%20transferencia%20de%20estilo%20es,%C3%A1s%C20im%C3%A1genes.>
- 3) <https://arxiv.org/ftp/arxiv/papers/2302/2302.09346.pdf>
- 4) <https://medium.com/@birla.deepak26/autoencoders-76bb49ae6a8f>
- 5) <https://repositorio.uchile.cl/bitstream/handle/2250/152952/Identificaci%C3%B3n-aut%C3%B3noma-de-fallas-en-sistemas-monitoreados-basado-en-redes-adversarias-generativas.pdf?sequence=1>