**Jefferson Sneyder Anaya Manrique 2210981**

**Jairo Andrés Gómez Molina 2185618**

**1. Aplicaciones de los Variational Autoencoders (VAE)**

Los **Variational Autoencoders** (VAE) son un tipo de autoencoder que combina técnicas de redes neuronales con principios de probabilidad para modelar datos complejos. Los VAEs han demostrado ser particularmente útiles en la generación de nuevos datos similares a los datos de entrenamiento. Aquí te presento dos aplicaciones destacadas:

**a) Generación de Imágenes Realistas y Modificación de Atributos**

Los VAEs se utilizan en la generación de imágenes realistas y en la edición de atributos específicos de las imágenes. Por ejemplo, se pueden aplicar VAEs para cambiar características de rostros como la expresión facial, el color del cabello, o incluso para agregar detalles como gafas o modificar la edad.

* **Ejemplo**: En investigaciones sobre imágenes de rostros humanos, se pueden entrenar VAEs con un conjunto de datos de rostros para aprender una representación latente. Luego, se pueden usar estos modelos para generar rostros completamente nuevos con diferentes atributos, o para modificar rasgos de rostros existentes de manera controlada.
* **Ventajas**: Esta técnica permite una edición flexible de imágenes y una generación realista de contenido visual, lo cual es útil en diseño gráfico, entretenimiento y aplicaciones de realidad aumentada.

**b) Síntesis de Música y Audio**

Los VAEs también se aplican en la generación de música y síntesis de audio. Mediante el aprendizaje de las representaciones latentes de señales de audio, un VAE puede generar nuevas secuencias de audio que imitan el estilo o las características de la música original.

* **Ejemplo**: Proyectos como **NSynth** de Google utilizan VAEs para crear nuevos sonidos combinando características de instrumentos existentes. Esto se logra modelando la representación latente de las formas de onda de audio y permitiendo la síntesis de sonidos completamente nuevos que nunca han sido grabados.
* **Ventajas**: Esta aplicación es útil para músicos y productores que buscan sonidos innovadores y únicos, y también en la creación de efectos de audio en videojuegos y aplicaciones de sonido envolvente.

**2. Aplicaciones de Redes Adversarias Generativas (GAN)**

Las **Redes Adversarias Generativas (GAN)** son una clase de modelos de aprendizaje profundo que han revolucionado la generación de datos sintéticos. Están compuestas por dos redes neuronales (generador y discriminador) que compiten entre sí, lo que da lugar a la generación de datos que pueden ser extremadamente realistas. Aquí se presentan dos aplicaciones notables:

**a) Generación de Imágenes de Alta Resolución y Hiperrealistas**

Una de las aplicaciones más destacadas de las GAN es la generación de imágenes de alta calidad que pueden imitar fotografías reales. Estas imágenes pueden representar rostros, paisajes, objetos y más, y a menudo son tan realistas que es difícil distinguirlas de imágenes auténticas.

* **Ejemplo**: El proyecto **"This Person Does Not Exist"** utiliza un modelo GAN para generar rostros humanos ficticios de alta resolución que no pertenecen a personas reales. Este tipo de tecnología se basa en variaciones de las GAN, como **StyleGAN**, que permite un control detallado sobre los atributos de las imágenes generadas.
* **Ventajas**: Esta aplicación es útil en la creación de contenido digital para marketing, cine, videojuegos y simulaciones, permitiendo la generación de imágenes sin necesidad de modelos o derechos de imagen.

**b) Mejora y Restauración de Imágenes**

Las GAN también se utilizan para mejorar la calidad de imágenes de baja resolución y para restaurar imágenes dañadas. Modelos como las **Super-Resolution GANs (SRGAN)** son capaces de aumentar la resolución de las imágenes y mejorar los detalles de manera notable.

* **Ejemplo**: En el campo de la fotografía y el cine, las GAN pueden usarse para convertir videos antiguos de baja resolución en versiones de alta definición. Otro ejemplo es la restauración de fotos antiguas que han sufrido deterioro, donde las GAN pueden completar y mejorar las áreas faltantes.
* **Ventajas**: Este uso es valioso para la preservación de patrimonio cultural y en aplicaciones médicas, donde se necesita mejorar la calidad de imágenes de diagnóstico.

**Diferencia entre un VAE y una GAN en la Generación de Información**

Aunque tanto los **Variational Autoencoders (VAE)** como las **Redes Adversarias Generativas (GAN)** se utilizan para la generación de datos, existen diferencias clave:

* **Arquitectura y Enfoque**:
  + **VAE**: Utilizan un enfoque basado en la codificación probabilística, donde el modelo aprende una distribución latente que permite generar muestras similares a los datos de entrenamiento. Son modelos más estables y más fáciles de entrenar, pero las imágenes generadas pueden no ser tan nítidas o detalladas como las de las GAN.
  + **GAN**: Utilizan dos redes (generador y discriminador) que compiten entre sí. Esta competencia mejora la calidad de las muestras generadas, que pueden ser más realistas y detalladas. Sin embargo, entrenar GAN es más desafiante debido a la naturaleza adversaria del modelo y puede requerir ajustes finos para evitar problemas como el *colapso de modo*.
* **Resultado de la Generación**:
  + **VAE**: Las imágenes generadas pueden presentar un ligero desenfoque y tienden a ser menos precisas debido a las aproximaciones probabilísticas.
  + **GAN**: Producen imágenes de mayor realismo y calidad visual, lo cual es ideal para aplicaciones donde la precisión visual es clave.

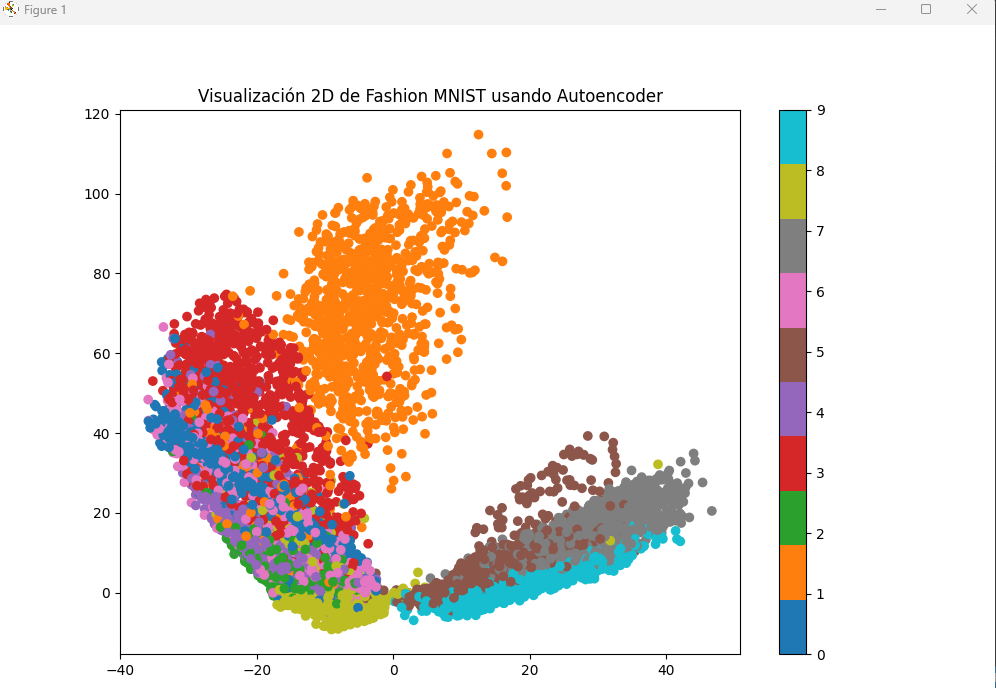
3. AutoencodersMNIST.py

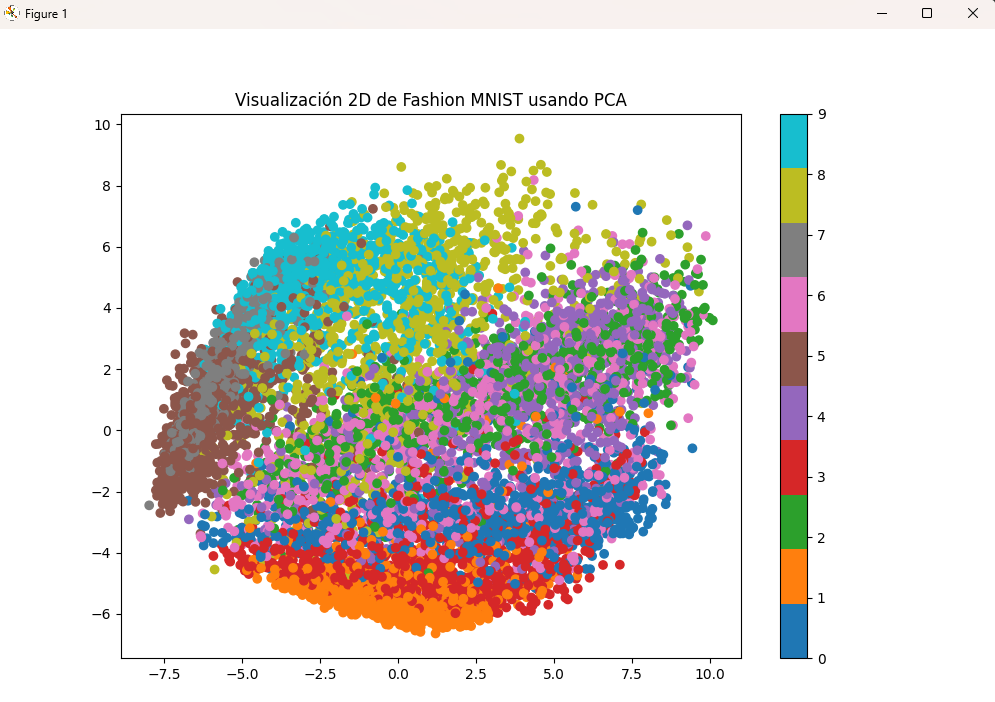
Resultado:

Texto

Descripción generada automáticamente

4. 2DimensionesMNIST.py

Resultados:  




Texto

Descripción generada automáticamente con confianza media

**Visualización y agrupamiento**:

En la gráfica del autoencoder (primera imagen), se observa una mejor separación y agrupación de las clases, lo que indica que la técnica de autoencoder ha aprendido una representación que discrimina mejor entre las categorías de Fashion MNIST.

En la gráfica de PCA (segunda imagen), los datos están más dispersos y no se ven patrones claros de agrupamiento para cada clase. Esto puede significar que PCA, al ser una técnica lineal, no captura tan bien las complejas relaciones no lineales de los datos como lo hace un autoencoder.

Las imágenes reconstruidas (tercera imagen) muestran ejemplos de las salidas del autoencoder. Si las imágenes son borrosas o poco definidas, podría indicar que el autoencoder necesita ajustes (como más entrenamiento, más neuronas, etc.), pero en general, se pueden ver algunos patrones representativos de la moda en ellas.

**La técnica de autoencoder ofrece una mejor representación en este caso para Fashion MNIST**, dado que captura relaciones no lineales y agrupa las clases de manera más clara en el espacio 2D comparado con PCA, que es una técnica de reducción lineal y no se adapta a datos con patrones complejos de manera tan efectiva.

5. AutoencodersApiladosMNIST.py

Resultados:

Texto

Descripción generada automáticamente

**Bibliografía:**

**Fuentes de documentación y bibliografía**:

**Documentación de TensorFlow**: <https://www.tensorflow.org/>

**Documentación de Git**: <https://git-scm.com/doc>

**Fashion MNIST Dataset**: <https://github.com/zalandoresearch/fashion-mnist>

**Participación de ChatGPT**:

**Corrección de errores ortográficos y gramaticales**: ChatGPT fue utilizado para revisar y corregir errores ortográficos y gramaticales en la documentación del proyecto.

**Estructuración y organización del contenido**: ChatGPT ayudó a organizar y estructurar el documento para hacerlo más claro y fácil de entender, explicando el código y los pasos necesarios para implementar un modelo de autoencoder y clasificador con la base de datos Fashion MNIST.

**Manual de buenas prácticas de programación**:

**PEP 8 – Style Guide for Python Code**: <https://peps.python.org/pep-0008/>

**Github:**

<https://github.com/jefryerson2003/Autoencoders>