

Prototipo de sistema predictor de secuencia de imágenes por medio de un Autoencoder

Trabajo Terminal No. _____

Alumno: Díaz Vital Héctor

Director: M. en C. Edmundo René Durán Camarillo

*e-mail: hectordiaz4iv7@gmail.com

Resumen: Este trabajo terminal implementará un autoencoder, el cual se define como una Red Neuronal Artificial (RNA) no supervisada que aprenderá a codificar y comprimir datos eficazmente, para después aprender a reconstruir datos a partir de los datos comprimidos teniendo como objetivo que se parezcan lo más posible a los datos originales. Para el caso particular de este Trabajo el autoencoder estará compuesto de un encoder Long Short Term Memory (LSTM), el cual permite recibir como entrada una secuencia de datos en este caso los valores de los píxeles de varias imágenes además de tener conexiones retroalimentadas. El siguiente componente es un decoder Long Short Term Memory (LSTM), el cual toma de entrada la salida del encoder y retorna una secuencia de datos en este caso píxeles que componen una imagen.

Palabras clave - Inteligencia Artificial, Redes Neuronales Artificiales (RNA), Ingeniería de software, Procesamiento de imágenes.

1. Introducción

Los Autoencoder (o autocodificador) han sido una herramienta muy útil en el procesamiento de imágenes, por ejemplo, son empleados para quitar el ruido, marcas de agua, o incluso realizar operaciones inversas a los filtros aplicados a una imagen. La razón de que sea posible obtener estos resultados radica en que los autocodificadores utilizan RNA que incluyen un cuello de botella dentro de la arquitectura, lo cual permite la posibilidad de agregar o ignorar elementos en el momento de pasar por la capa que contiene menor número de nodos a la capa con mayor número de nodos.

De forma más específica en este Trabajo se realizarán los siguientes pasos para diseñar el autocodificador: primero, las imágenes de entrada pasarán por un preprocesamiento, en el cual se representarán las imágenes como un vector que tendrá valores según el valor de los píxeles, después entrará en la fase de encoding la cual se representa como una reducción dimensional del vector implicado que dado un vector de datos m -dimensional \mathbf{x} , puede ser representado por un vector l -dimensional de componentes principales, como se ilustra en la Ec. (1)

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_l \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1^T \\ \mathbf{q}_2^T \\ \vdots \\ \mathbf{q}_l^T \end{bmatrix} \mathbf{x}, \quad l \leq m \quad (1)$$

donde \mathbf{q}_i es el vector propio i -ésimo de la matriz de correlación m -por- m

$$R = E[\mathbf{x}\mathbf{x}^T] \quad (2)$$

y a_i , el componente i -ésimo del vector \mathbf{a} , es la proyección del vector de datos \mathbf{x} sobre el vector propio i -ésimo \mathbf{q}_i . Si $l = m$, entonces el nuevo vector \mathbf{a} es una versión rotada del vector de datos original \mathbf{x} ; la verdadera diferencia entre ellos es que \mathbf{a} tiene componentes no correlacionados, mientras que \mathbf{x} no. Si $l < m$, entonces solo se conserva un subconjunto de los vectores propios, lo que hace que la representación de los datos sea aproximada. En este último caso, hablamos de reducción de dimensionalidad [1].

Al vector resultante le aplicaremos un proceso de decoding, el cual definiremos como dado el vector de componentes principales \mathbf{a} , se establece que el vector de datos original \mathbf{x} puede ser reconstruido por una combinación lineal de los vectores propios, como lo muestra la Ec. 3 [2].

$$\hat{x} = \sum_{i=1}^l a_i q_i, \quad l \leq m \quad (3)$$

En la reconstrucción de datos es en donde se le enseña a la RNA las coincidencias con la imagen siguiente de una secuencia.

Hay muchos tipos de autocodificadores, según su objetivo de uso y su arquitectura, para este trabajo terminal se utilizará un autocodificador Long Short Term Memory (o Memoria a Largo y Corto Plazo) esto implica que se conformará por un codificador con Memoria a Largo y Corto Plazo y un decoder con Memoria a Largo y Corto Plazo. Este tipo de autocodificadores están dedicados a resolver problemas del tipo “sequence-to-sequence” este tipo de problemas pueden llegar a ser complejos por el hecho de que la cantidad de datos de entrada y salida de nuestro puede variar. En este caso la cantidad de datos de salida de nuestro autocodificador dependerá directamente del tamaño de nuestras imágenes de entrada.

SOFTWARE	DESCRIPCIÓN	Algoritmos y/o tipo redes neuronales usadas
Selección de los colores esenciales de una imagen digital	Este trabajo propone un algoritmo no supervisado para la cuantificación del color en las imágenes digitales. El propósito principal es que sobre la base de los colores que originalmente están presentes en la imagen, se obtenga una nueva imagen que cuente con un mínimo de los colores que denominamos esenciales que permitan conservar los objetos que componen la imagen original. [3]	Algoritmo de popularidad Algoritmo k-medias Algoritmo k-medias difuso Algoritmo de popularidad Algoritmo de corte de mediana Algoritmo octree Algoritmo de componentes principales
Un compresor de imágenes híbrido en capas con técnica de aprendizaje profundo	En este trabajo propone un sistema de compresión formado por un códec intra VVC como capa base y un códec residual basado en el aprendizaje como capa de mejora. Este último tiene como objetivo refinar la calidad de la capa base mediante el envío de una señal residual latente. En particular, se emplea un módulo de atención guiada por la capa base para centrar la extracción residual en áreas críticas de alta frecuencia. Para reconstruir la imagen, esta señal residual latente se combina con la salida de la capa base de manera no lineal mediante un sintetizador basado en redes neuronales.[4]	Error cuadrado medio Similitud estructural multiescala Codificación aritmética binaria de mapas de entidades Red neuronal autoencoder con capas de convolución simétricas Red neuronal con capas de bloques residuales Red neuronal encoder de entropía
Arquitectura de Autoencoder para predecir gráficos dinámicos	En este trabajo se desarrolla e implementa un autoencoder predictivo para datos gráficos temporales secuenciales (es decir, datos procesados linealmente que	Red neuronal autoencoder Long Short Term Entrenamiento para datos secuenciales Entrenamiento para datos no

	representan el crecimiento del gráfico) y no secuenciales. [5]	secuenciales
Prototipo de sistema predictor de secuencia de imágenes por medio de un Autoencoder	Este trabajo propone implementar un autoencoder Long Short Term Memory el cual permite recibir como entrada una secuencia de datos en este caso los valores de los pixeles de varias imágenes y retorna una secuencia de datos que representan pixeles que componen una imagen que resulta ser la predicción secuencial correspondiente de las imágenes de entrada.	Red neuronal autoencoder Long Short Term Vectorización de una imagen Entrenamiento para datos secuenciales Combinación lineal de vectores

Tabla 1.- Comparativa del estado del arte

2. Objetivo

Objetivo General:

- Desarrollar un prototipo de sistema que permita predecir a partir de una secuencia de imágenes, otra imagen correspondiente a esa secuencia

Objetivos específicos:

- Diseño e implementación de un autocodificador Long Short Term Memory que sea capaz de predecir adecuadamente una imagen a partir de una secuencia de imágenes n-dimensionales en escala de grises
- Manual de usuario
- Reporte técnico
- Manual de instalación

3. Justificación

Los autocodificadores tienen múltiples utilidades en distintos sectores. Algunas aplicaciones de ellos son las siguientes, pero existen muchísimas más: [6]

- Compresión de información: la principal característica de este tipo de algoritmo es que son capaces de extraer las características más importantes de la información y crear una nueva representación de esta en una dimensión reducida.
- Detección de fraudes: los autocodificadores han sido utilizados por empresas como PayPal para construir sistemas de detección de fraudes extrayendo las características clave que determinan si una cierta transacción es fraudulenta o no.
- Generación de imágenes: otra aplicación posible es la generación de imágenes. Esto puede ser muy útil para tareas de diseño y para la generación de nuevos datasets de imágenes de libre distribución.

Dicho esto, puede decirse que este es un trabajo que aporta una base a la realización de un predictor de fotogramas faltantes de videos dañados, además de que permitirá diversificar los alcances en el campo de las predicciones por medio de videos. Así es que durante los últimos años se ha resaltado la relevancia de los algoritmos predictivos fundamentados en deep learning, un ejemplo de esto podría ser la herramienta implementada por los automóviles TESLA los cuales tienen incorporado un analizador de fotogramas, el cual se encarga primero de detectar objetos en una imagen y predecir el movimiento de dichos objetos y reaccionar a ello, lo que resulta en una conducción automática.[7]

En el presente trabajo terminal se diseñará un autocodificador que podrá ser utilizado para sistemas predictivos de imágenes más complejos como por ejemplo, poder obtener otra secuencia de imágenes lo que resultaría más específicamente en la predicción de fotogramas de un video. Esto resultaría muy útil para recomponer videos con fotogramas corruptos o incompletos. También podría usarse como una base general para predecir fotogramas de cierto tipo de videos específicos, como en el ejemplo del automóvil TESLA.

Independientemente del uso en el ámbito de los videos, también sería un buen aporte para analizar por ejemplo obras de arte y categorizar el orden en que fueron pintadas y por cual pintor, esto requeriría de una red neuronal adicional del tipo clasificador.

Considero que este aporte ayudará mucho a establecer un espacio más cómodo para experimentar y estimular el desarrollo de autocodificadores para analizar secuencia de datos, ya que lamentablemente no tiene una gran popularidad el uso de esta tecnología a comparación de otras y creo firmemente que es una buena herramienta que amplía el abanico de posibilidades a utilizar para resolver problemas utilizando inteligencia artificial

Este trabajo planea evidenciar e implementar los conocimientos obtenidos, durante el sendero educativo que se ha recorrido durante la carrera de ingeniería en sistemas computacionales en el Instituto Politécnico Nacional, en materias como ingeniería de software, análisis de algoritmos, algoritmos, algebra lineal, programacion de sistemas avanzados de cómputo e inteligencia artificial, específicamente al desarrollo de un autocodificador, rama la cual está actualmente en un constante y trepidante cambio e innovación que genera mucho interés para los amantes de la tecnologia y programacion ya que resulta ser una herramienta útil y cada vez más relevante para la resolución de problemas cada vez más complejos.

4. Productos o resultados esperados

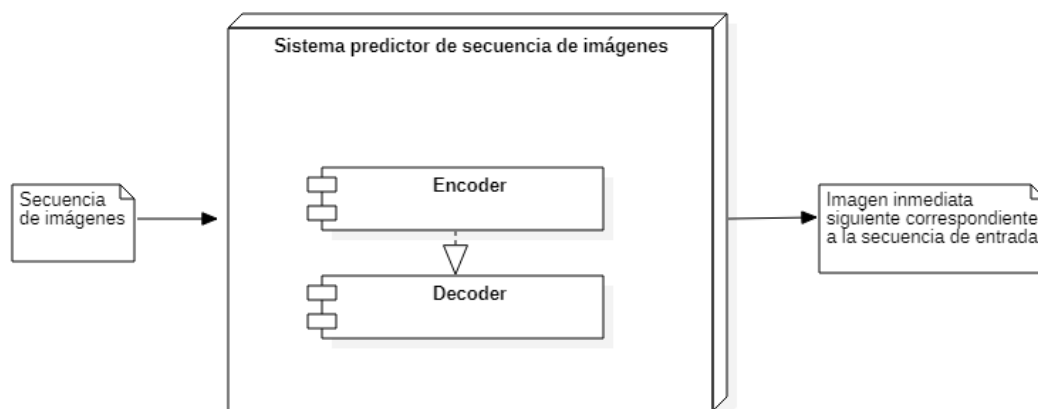


Figura 1.- Arquitectura propuesta del software

Se entregarán los siguientes productos:

- Manual de usuario
- Reporte técnico
- El código del sistema computacional, el cual incluye:
 - Encoder
 - Decoder
 - Imágenes de prueba con sus respectivas predicciones
 - Motor de aprendizaje y evaluación

5. Metodología

Modelo de Prototipos. También conocido como modelo de desarrollo evolutivo, se inicia con la definición de los objetivos globales para el software, luego se identifican los requisitos conocidos y las áreas del esquema en donde es necesaria más definición. Este modelo se utiliza para dar al usuario una vista preliminar de parte del software. Este modelo es básicamente prueba y error ya que si al usuario no le gusta una parte del prototipo significa que la prueba falló por lo cual se debe corregir el error que se tenga hasta que el usuario quede satisfecho. Además el prototipo debe ser construido en poco tiempo, usando los programas adecuados y no se debe utilizar mucho dinero pues a partir de que este sea aprobado nosotros podemos iniciar el verdadero

desarrollo del software. Pero eso si al construir el prototipo nos asegura que nuestro software sea de mejor calidad, además de que su interfaz sea de agrado para el usuario. Un prototipo podrá ser construido sólo si con el software es posible experimentar.[8]

Esta metodología nos permite una gran flexibilidad y adaptación y considero que es el idóneo para desarrollar el sistema propuesto por este trabajo. Para ilustrar el procedimiento de esta metodología podemos recurrir a la Figura 2: Proceso gráfico del modelo de prototipos.

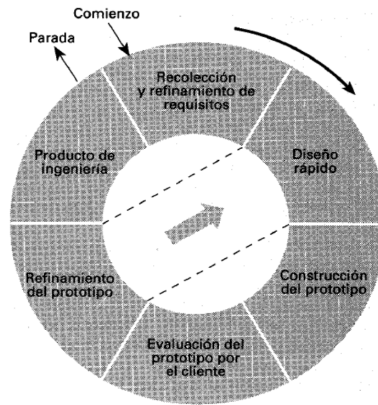


Figura 2: Proceso gráfico del modelo de prototipos

Algunas ventajas a resaltar de esta metodología son:

- No modifica el flujo del ciclo de vida
- Reduce el riesgo de construir productos que no satisfagan las necesidades de los usuarios
- Reduce costo y aumenta la probabilidad de éxito
- Exige disponer de las herramientas adecuadas
- También ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción humano-máquina. [9]

6. Cronograma

Ver el Anexo 1.

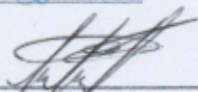
7. Referencias

- [1] [2] S. Haykin, "Neural Networks and Learning Machines," 3rd ed. New Jersey, NJ, USA: Ed. Pearson 2008, pp. 412-413.
- [3] A. M. Sergio, "Selección de los colores esenciales de una imagen digital," M.S. thesis, Instituto Politécnico Nacional Univ., Ciudad de México, México, 2008.
- [4] W.-C. Lee, C.-P. Chang, W.-H. Peng, and H.-M. Hang, "A Hybrid Layered Image Compressor with Deep-Learning Technique," presented at the 22nd International Workshop on Multimedia Signal Processing, Tampere, Finland, 21-24 Sept. 2020.
- [5] I. Costley and M. O'Gara, "Autoencoder Architecture for Predicting Dynamic Graphs," presented at the IEEE MIT Undergraduate Research Technology Conference, Cambridge, MA, USA, 5-7 Oct. 2018.
- [6] R. Cañadas, "Qué son los autoencoders," *Autoencoders | Qué son, Arquitectura y sus Aplicaciones*, Agosto 2021. [Online]. <https://abdatum.com/machine-learning/autoencoders> [Accessed: Oct. 26, 2021].
- [7] "Artificial Intelligence & Autopilots," *TESLA*, 2021. [Online] Available: <https://www.tesla.com/AI> [accessed Oct. 23, 2021]
- [8] [9] "Modelo de prototipos," *Ecures.cu*, 2019. [Online] Available: https://www.ecured.cu/Modelo_de_prototipos [Accessed: Oct. 20, 2021]

8. Alumnos y directores

Héctor Díaz Vital. Alumno de la carrera de Ing. en Sistemas Computacionales en ESCOM, Especialidad de Sistemas Computacionales, Boleta: 2016630095 , Tel. 5525149372, email: hectordiaz4iv7@gmail.com

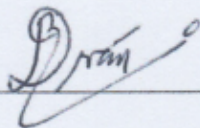
Firma: _____



Edmundo René Durán Camarillo. Docente de la ESCOM-IPN; ha dirigido trabajos terminales durante 25 años.

Áreas de Interés: Automatización, Inteligencia Artificial, Sistemas Neurodifusos sistemas expertos. Datos de contacto: Extensión dentro del IPN 52037, Cel. 55 22721080; correo: eduranc@ipn.mx.

Firma: _____



CARÁCTER: Confidencial
FUNDAMENTO LEGAL: Artículo 11 Fracc. V y Artículos 108, 113 y 117 de la Ley Federal de Transparencia y Acceso a la Información Pública.
PARTES CONFIDENCIALES: Número de boleta y teléfono

TT No.:

Titulo del TT: Prototipo de sistema predictor de secuencia de imágenes por medio de una red neuronal tipo autoencoder

[illegible]

[illegible]