

---

## Proyecto 2

### Redes Neuronales

---

Fecha de asignación: 23 de Octubre, 2025  
Grupos: 3 personas

Fecha de entrega: 11 de Noviembre, 2025  
Profesor: Jason Leiton Jiménez

---

## 1. Objetivo

Implementar redes neuronales con el fin de solucionar un problema de clasificación utilizando un set de datos de imágenes público.

## 2. Motivación

En la lucha contra la pandemia, inicialmente surgieron formas de observar el daño pulmonar que causa el Covid19 en imágenes de rayos X. Este daño es distinto al causado por pulmonía, aunque similar. Para el ojo entrenado, puede que sea más sencillo saber las diferencias, pero en general puede ser útil tener un sistema automático que ayude a detectar esos patrones en las fotos, que sirva de apoyo a los médicos e incluso a la población en general.

Este proyecto se enfoca en un set de datos de rayos X tomados de personas con Covid19, con pulmonía, sanos, y una cuarta categoría. Es un problema supervisado, ya que tenemos los labels de cada una de las 4 categorías.

Una de las grandes ventajas que tiene las redes neuronales es la capacidad de modelar problemas que no son lineales y que tienen gran cantidad de dimensiones, como lo son las imágenes. Por esta razón es que se implementará el proyecto utilizando diferentes redes neuronales.

## 3. Descripción General

El data set a utilizar en este proyecto es: <https://www.kaggle.com/datasets/preetviradiya/covid19-radiography-dataset>. El cual consiste en un conjunto de imágenes de rayos X del torax de las personas. Cada imagen tiene una etiqueta asociada, la cual corresponde al tipo enfermedad o si está sano (Covid, neumonía, sano y opacidad pulmonar). En la imagen 1 se muestra un ejemplo de imagen.



Figura 1: Ejemplo de imagen de Data Set

El proyecto consiste utilizar diferentes arquitecturas de redes neuronales para crear un modelo capaz de indicar si una persona tiene covid o neumonía a partir de los rayos X de torax.

Se debe de realizar 2 tipos de redes neuronales las cuales son MLP y CNN. Para este proyecto, es de manera obligatoria realizar todo el análisis de feature engineering

### 3.1. MLP (Perceptron multi capa)

La red neuronal más común es la red que consiste en una serie de capas, las cuales se conectan entre sí de manera total. En esta parte cada grupo deberá proponer la arquitectura que mejores resultados arroja. Se debe de establecer al menos tres diseños de una red. Esta red debe utilizarse en conjunto con algún feature extractor a elección de cada grupo. La idea de esta sección es que se debe hacer una comparación de resultados (métricas vistas y tiempo de entrenamiento) entre la red con el feature extractor y sin este.

El feature extractor es de elección libre. Básicamente algoritmos estilo Local Binary Patterns, Histogramas de Color, SIFT, ORB, entre otros, permiten tomar la imagen y convertirla en un vector de menor dimensionalidad, capturando lo que cada algoritmo cree que son features más representativos. Estos features extractors no son entrenables. Es decir, este preprocesamiento solo se ejecuta una única vez, **NO** se debe de ejecutar cada vez que se requiera probar el modelo.

### 3.2. Redes Convolucionales

Las redes de convolución suelen ser más eficientes ya que contienen menos parámetros que aprender, pueden ser entrenadas en menos tiempo y necesitan menos espacio en memoria. Hay muchos modelos de redes de convolución, algunos MUY pesados otros más sencillos. Se debe buscar uno que se ejecute en el ambiente de trabajo de cada grupo, o bien, crear su propia red de convolución más pequeña (pueden utilizar alguna).

Se debe comparar ejecutar la CNN con las imágenes crudas, y también con un filtro llamado Bilateral Filter. Este filtro toma en cuenta el vecindario de cada pixel para reemplazar su valor, con un efecto de suavidad y de preservación de bordes. Existen muchas implementaciones, incluyendo la de OpenCV. Se recomienda hacer el filtrado de las imágenes offline (solo se ejecuta una vez), no durante el entrenamiento, y guardarlas ya filtradas para evitar tener que filtrar una y otra vez.

### 3.3. Visualización de Mapas de Calor

Se debe visualizar, una vez entrenada alguna de las redes, cuáles son las zonas de la imagen que más influencia tuvieron en la toma de decisiones respecto a la clasificación final. Se recomienda usar Saliency Maps, aunque es posible usar otras técnicas. En la figura 2 se muestra un ejemplo de cómo se espera que se muestren las imágenes de pruebas.

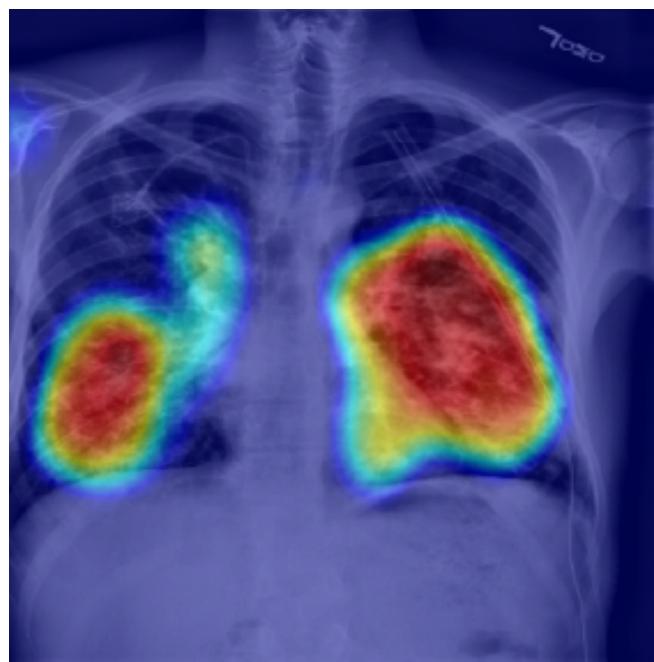


Figura 2: Ejemplo de mapa de calor

### 3.4. Métricas

Se debe reportar para todos los casos el Accuracy por clase, así como las otras métricas derivadas de la matriz de confusión. Es importante destacar que se deben de obtener las métricas del test, sin embargo, las de entrenamiento son necesarias para referirse al overfitting y underfitting.

Las comparaciones entre las métricas deben ser por medio de tablas, en donde sea sencillo la lectura y la evaluación por ejecución.

Se espera que se respondan de manera concreta las siguientes preguntas:

- ¿Cuál es el tamaño de los modelos?
- ¿Cuál es la arquitectura (cuales y cuantas capas se utilizaron)?
- ¿Qué ventajas tiene utilizar un feature extractor? referirse a las métricas y tiempos
- ¿Qué modelo se entrena más rápido?
- ¿Cuáles secciones de la imagen son las más importantes para la elección?
- ¿Cuál fue el mejor modelo?

## 4. Requerimientos técnicos

Las redes neuronales deben implementarse en Pytorch. Se recomienda el uso de GPUs en Google Colab, subiendo el set de datos en Google Drive. De tener GPU propio, todavía mejor. Otra opción para obtener GPUs es usar el cluster Kabré de CeNAT, o bien en caso de tener la posibilidad, Azure o AWS son opciones viables.

## 5. Documentación- Estilo IEEE-Trans

- Introducción: Teoría necesaria, breve descripción del proyecto y qué es lo que se espera en el escrito.
- Detalles del diseño del programa desarrollo. En esta sección se espera observar un diagrama de flujo de cada algoritmo, así como la elección de los parámetros a utilizar para cada problema. También los diagramas de arquitecturas de las redes neuronales.
- Resultados y análisis. Se espera ver todas las métricas así como la generalización de las mismas, aquí se debe mencionar los problemas de los set, si hubo overfitting, underfitting, cuál fue el mejor, las razones de los mejores y peores resultados, mencionar comparaciones de resultados con parámetros y cualquier otro análisis que sea relevante

- Conclusiones
- Referencias

## 6. Entregables

1. Notebook: Un solo notebook(.ipynb) con el código y con los mejores resultados de cada modelo.
2. Paper: PDF con el contenido de la documentación

## 7. Fecha de entrega y revisión

11 de noviembre 9:30. Revisiones en horario de clase.

## 8. Evaluación

- MLP (experimentos y análisis) 15 %
- MLP con feature extractor (experimentos y análisis) 15 %
- CNN sin filtros (experimentos y análisis) 15 %
- CNN con filtros (experimentos y análisis) 15 %
- Visualización de mapas de calor 10 %
- Paper 30 %