

Proyecto final

Análisis de imágenes médicas

Para el curso de: Programador IA

Autor: José Agustín Ortiz

Índice

1. Objetivos	3
2. Resumen	4
3. Marco Teórico	5
3.1. Redes convolucionales	5
3.1.1. Capas convolucionales	5
3.1.2. Capas de agrupación	6
3.1.3. Capa totalmente conectada	6
3.2. Función Rectificada Uniforme(RELU)	6
3.3. Función de pérdida	7
3.3.1. Función de entropía cruzada binaria	8
3.4. Adam de algoritmo de optimización	8
4. Recursos empleados	9
5. Desarrollo	10
6. Análisis y discusión de datos	11
6.1. Arquitectura del modelo	11
6.2. Entrenamiento	12
7. Conclusiones	14

1. Objetivos

Aplicar inteligencia artificial para asistir en la detección de anomalías médicas mediante redes convolucionales.

2. Resumen

Este documento fue creado con el objetivo de explorar formas de aplicar la inteligencia artificial y la programación en el campo de la medicina. El propósito principal es no solo facilitar la detección de anomalías en imágenes de rayos X y ofrecer una visualización clara del proceso. En este contexto, utilizaremos una base de datos que incluye registros de casos de COVID-19, aunque el concepto puede generalizarse para abarcar otros tipos de enfermedades o condiciones médicas.

3. Marco Teórico

3.1. Redes convolucionales

El concepto principal que abarca este proceso es el de redes neuronales convolucionales, un subconjunto de machine learning y concepto clave en algoritmos de deep learning. Se caracteriza por su rendimiento en entradas de imagen, voz y señales de audios.

En su arquitectura las primeras capas se centran en características simples, tales como colores o bordes, pero a medida que la imagen avanza por la estructura, la red puede reconocer elementos más complejos. Para posteriormente asignarle a la imagen a alguna categoría, descripción o etiqueta.

Las redes neuronales convolucionales se componen de tres tipos de capas: Las capas convolucionales, la capa de agrupación y la capa totalmente conectada.

3.1.1. Capas convolucionales

Las capas convolucionales es donde se realizan la mayoría de los cálculos, requiriendo datos de entrada, un filtro y un mapa de características. Los datos de entrada son un tensor de una determinada altura, anchada y profundidad. La profundidad no necesariamente puede ser de tres como con la composición RGB en una imagen, también puede ser de uno como se vio en este proyecto que usamos radiografías en escalas de grises. Además hay un detector de características, el Kernel, que se mueve por los campos receptivos de la imagen para comprobar si la característica se encuentra presente, se desplaza por medio de saltos con una distancia determinada en pixeles por el Stride. El Padding controla cómo se manejan los bordes de la imagen durante la convolución. Podemos agrupar varias capas de convolución adicionales posteriormente para ir identificando patrones desde un bajo nivel hasta un alto nivel, aumentando la complejidad de las características extraídas.



Figura 1: Representación de la jerarquía de características en redes neuronales convolucionales. Fuente: IBM

3.1.2. Capas de agrupación

Las capas de agrupación permiten reducir la dimensión mediante la reducción del número de parámetros de entrada. Barre toda la entrada con un filtro pero no tiene ningún peso, aplicando una función de agregación dentro de los valores del campo receptivo y llena así la matriz de salida. Hay dos tipos principales de agrupación: Agrupación máxima, que selecciona el pixel más alto a medida que el filtro recorre la entrada y la agrupación media que calcula el valor promedio dentro del campo receptivo, el correspondiente pixel de valor más alto o el valor promedio serán enviados a la matriz de salida.

3.1.3. Capa totalmente conectada

Capa totalmente conectada, cada nodo de la capa de salida sí está conectado directamente a un nodo de la capa anterior, su rol consiste en la clasificación de las características extraídas de las capas anteriores y sus diferentes filtros. Usan funciones de activación softmax para clasificar adecuadamente las entradas y generar una probabilidad entre 0 y 1.

3.2. Función Rectificada Uniforme(RELU)

Es una función definida de tal forma en que el valor de entrada x es 0 si x es negativo o $f(x) = x$ si x es positivo.

$$f(x) = \max(0, x)$$



Figura 2: Gráfico de la función RELU extraído del sitio web: Jacar

Las ventajas de la función Unidad Rectificada Uniforme(RELU) es que permite realizar sus cálculos sin mucho poder de procesamiento y es menos propensa al desvanecimiento del gradiente, el cual ocurre si la derivada de la función de activación se acercan a valores muy pequeños y dificultad la actualización de los pesos. Tampoco sufre de saturación de la activación como otras funciones de activación, como pueden ser la sigmoide o la tangente hiperbólica.

3.3. Función de pérdida

Una función de pérdida consiste en una función la cual su maximización o minimización represente el objetivo del entrenamiento del modelo. Rastrea el grado de error de las salidas de un modelo de IA. Esto se realiza a partir de cuantificar una diferencia del valor previsto por el modelo y el valor real. Si las predicciones son precisas entonces la pérdida será pequeña. De la misma forma si las predicciones no son cercanas, la pérdida será grande.

3.3.1. Función de entropía cruzada binaria

Este tipo de función de pérdida se utiliza para la clasificación binaria y genera valores entre 0 y 1, siendo 0,5 la máxima incertidumbre o entropía. Pese a ello, los valores reales para las predicciones serán 0 o 1, ya que deben representar a nada más una de las categorías teóricamente. Minimizar la pérdida de entropía cruzada binaria implica no solo penalizar las predicciones incorrectas, sino también penalizar las predicciones con baja certeza. La pérdida promedio en un conjunto completo de n ejemplos de entrenamiento se calcula como

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \left(y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \right) \quad (1)$$

3.4. Adam de algoritmo de optimización

Adam, Adaptive Moment Estimation, es un algoritmo de optimización que se usará para mejorar el proceso de aprendizaje del modelo, haciendo uso de una estimación del momento y la magnitud de los gradientes para actualizar los parámetros del modelo en cada iteración. Esto nos ayudará en el ajuste del modelo.

4. Recursos empleados

Hice uso de tres librerías principales durante el desarrollo:

- Matplotlib: Para la generación de gráficos y la visualización de imágenes.
- PyTorch: Por sus módulos que facilitaron la construcción y entrenamiento del modelo.
- NumPy: Para el manejo de arreglos.

Para almacenar la base de datos extraída de Kaggle[1], utilicé Google Drive. El entrenamiento del modelo y la ejecución del código se realizaron utilizando la GPU y CPU proporcionadas por la nube de Google Colab.

5. Desarrollo

Este documento surgió más que nada como una propuesta de como aplicar conceptos de Inteligencia Artificial para asistir al sector de salud. Un artículo que analizó esto fue el de un sitio especializado en cuestiones legales que trató acerca de como la fatiga médica y el agotamiento pueden causar errores médicos [2], algo muy característico pero problemático del rubro. Además, a lo largo del curso se discutió el potencial de la inteligencia artificial para actuar no como un reemplazo, sino como un complemento y herramienta en diversas actividades para mejorar la calidad de vida de la humanidad, permitiendo entrenarla para actividades más mecánicas teniendo un alto grado de precisión en sus análisis, de esa forma podría actuar como un asistente para el profesional en la detección de anomalías.

La idea es desarrollar herramientas tecnológicas que sirvan como apoyo para los médicos, ayudándolos a tomar decisiones informadas y a mitigar posibles errores que puedan tener consecuencias legales, psicológicas o permanentes en la salud de los pacientes. Es importante enfatizar que esto no implica responsabilizar al médico, ya que estos errores son muchas veces consecuencia de las condiciones laborales extremas a las que están sometidos.

Es bastante común que los médicos trabajen hasta 80 horas semanales, lo cual por el desgaste físico y emocional, impacta negativamente en sus funciones cognitivas, capacidad de memoria y toma de decisiones. Además, muchas de sus tareas se extienden fuera del entorno laboral, intensificando el agotamiento.

La raíz del problema no está en los profesionales de la salud, sino en los organismos y sistemas encargados de la gestión y regulación de las prácticas laborales, que implementan dinámicas perjudiciales tanto para los médicos como para los pacientes.

Proveerles herramientas de soporte basadas en tecnología no solo podría mejorar la calidad del diagnóstico y la atención, sino también reducir la carga mental de los profesionales. Esto permitiría un entorno de trabajo más seguro y humano, beneficiando a todas las partes involucradas.

Puntualmente como demostración y ejemplo confeccioné un modelo que a partir de radiografías puede identificar y distinguir si la placa del tórax corresponde o no a un paciente con SARS-CoV-2.

Aunque el modelo es efectivo para detectar SARS-CoV-2 en radiografías, su desempeño podría mejorar con una base de datos más amplia y diversa. Además, se podrían explorar arquitecturas más avanzadas como ResNet o DenseNet para aumentar la precisión.

6. Análisis y discusión de datos

6.1. Arquitectura del modelo

```
import torch
import torch.nn as nn
import torch.nn.functional as F
from torch.utils.data import Dataset

class SimpleNetwork(nn.Module):
    def __init__(self):
        super(SimpleNetwork, self).__init__()

        self.conv1 = nn.Conv2d(in_channels=1,
                                out_channels=16, kernel_size=5, stride=1,
                                padding=1)
        self.bn1 = nn.BatchNorm2d(16)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(in_channels=16,
                                out_channels=32, kernel_size=5, stride=1,
                                padding=1)
        self.bn2 = nn.BatchNorm2d(32)

        # Capa completamente conectada ajustada
        self.fc1 = nn.Linear(32 * 54 * 54, 1)

    def forward(self, x):
        x = F.relu(self.bn1(self.conv1(x)))
        x = self.pool(x)
        x = F.relu(self.bn2(self.conv2(x)))
        x = self.pool(x)

        # Aplanar y pasar por la capa completamente conectada
        x = x.view(x.size(0), -1)
        x = torch.sigmoid(self.fc1(x))
        return x

# Crea el modelo
model = SimpleNetwork()
```

6.2. Entrenamiento

Este modelo simple de redes neuronales convolucionales fue diseñado considerando las características previamente descritas en el marco teórico, con una arquitectura enfocada en el reconocimiento de patrones específicos presentes en las imágenes asignadas.

Su propósito es principalmente explicativo, permitiendo comprender los fundamentos de este tipo de redes sin un costo computacional elevado. Para su entrenamiento imágenes de 224x224, con el objetivo de poder detectar rastros de neumonía en placas de rayos x correspondientes a pacientes con y sin SARS-CoV-2. Estas imágenes fueron clasificadas con etiquetas correspondientes, asignando '1' a las radiografías de pacientes enfermos y '0' a las de pacientes sanos.

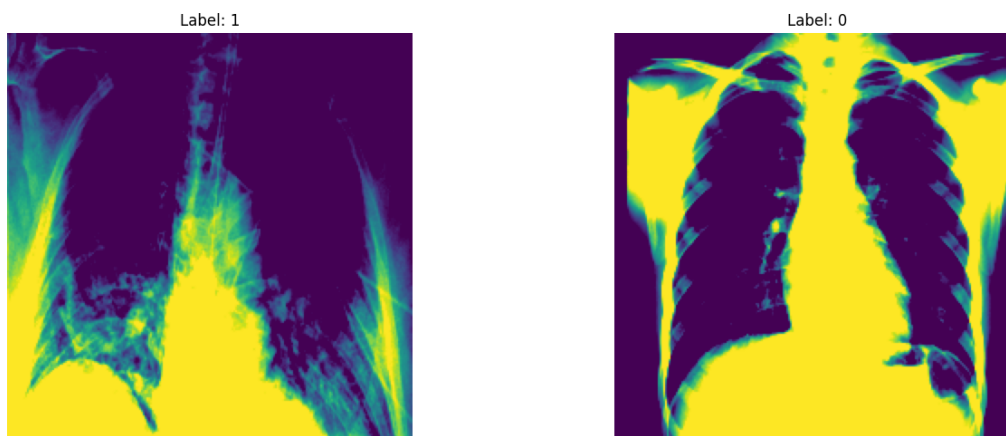


Figura 3: Ejemplo de radiografías de tórax de diferentes pacientes: uno diagnosticado como enfermo (*label 1*) y otro como sano (*label 0*).

Durante su entrenamiento se generó simultáneamente un gráfico para observar la evolución de la precisión del modelo durante el entrenamiento y durante las pruebas. Además se monitorearon los valores de la función de pérdida, lo cual permitió analizar el desempeño del modelo, todo a medida que iban avanzando las épocas.

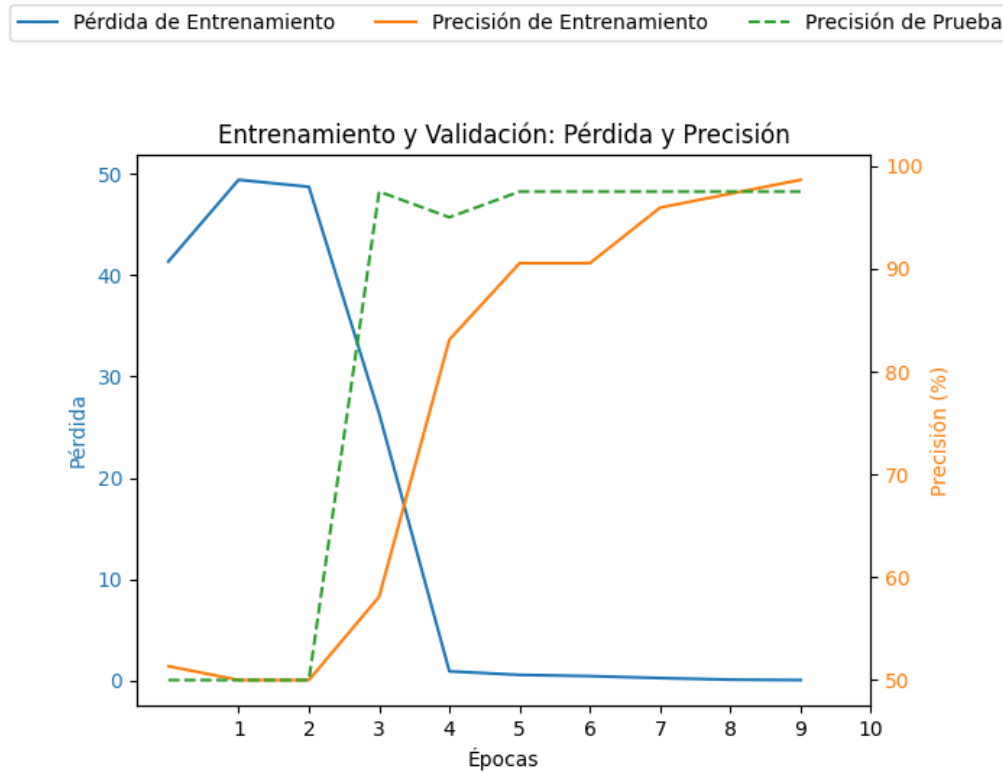


Figura 4: Gráfico que muestra la evolución de la precisión y la función de pérdida durante el entrenamiento y las pruebas.

Se puede apreciar como a lo largo de la evolución del modelo su desempeño fue mayor debido al aumento de la precisión tanto en el entrenamiento como en las pruebas, al igual que decayó el valor de perdida. Esto nos da una precisión final en el conjunto de prueba de 97,50 %.

Este tipo de programas para detección de anomalías en placas radiográficas no es algo nuevo, pero es un ejemplo de como podría implementarse las redes neuronales convolucionales en el sector médico.

7. Conclusiones

De la misma forma que se realizó esta implementación, uno puede generalizar este concepto no solamente para el caso de SARS-CoV-2, sino también para otro tipo de diagnósticos o situaciones que requieren la identificación de anomalías en imágenes, al igual que se pueden armar sistemas de clasificación todavía más complejos. Una herramienta así podría permitir al médico tener un salvavidas al momento de evaluar y evitar errores causados por el agotamiento.

También cabe resaltar que este tipo de modelos si se permite una mayor capacidad computacional, se pueden realizar modelos de inteligencia artificial de mayor complejidad integrando arquitecturas más avanzadas. Estas herramientas no son una solución al problema de raíz, pero sí puede actuar de apoyo en entornos donde el personal médico se encuentre bajo presión o cansancio, contribuyendo a mejorar la precisión diagnóstica y a colaborar en la atención al paciente.

Referencias

- [1] Wei Hao Khoong. Covid-19 xray dataset (train test sets)o. [Online]. Available: <https://www.kaggle.com/datasets/khoongweihao/covid19-xray-dataset-train-test-sets>
- [2] P. A. Salvi. La fatiga médica puede causar errores médicos. [Online]. Available: <https://www.salvilaw.com/es/negligencia-medica/fatiga-medica/>
- [3] IBM. Convolutional neural networks. [Online]. Available: <https://www.ibm.com/es-es/topics/convolutional-neural-networks>
- [4] Jacar. La función unidad rectificada uniforme (relu): Una herramienta esencial para el aprendizaje profundo. [Online]. Available: <https://jacar.es/la-funcion-unidad-rectificada-uniforme-relu-una-herramienta-esencial-para-el-aprendizaje-profundo/>
- [5] IBM. Función de pérdida. [Online]. Available: <https://www.ibm.com/es-es/think/topics/loss-function>
- [6] InteractiveChaos. Adam. [Online]. Available: <https://interactivechaos.com/es/manual/tutorial-de-machine-learning/adam>