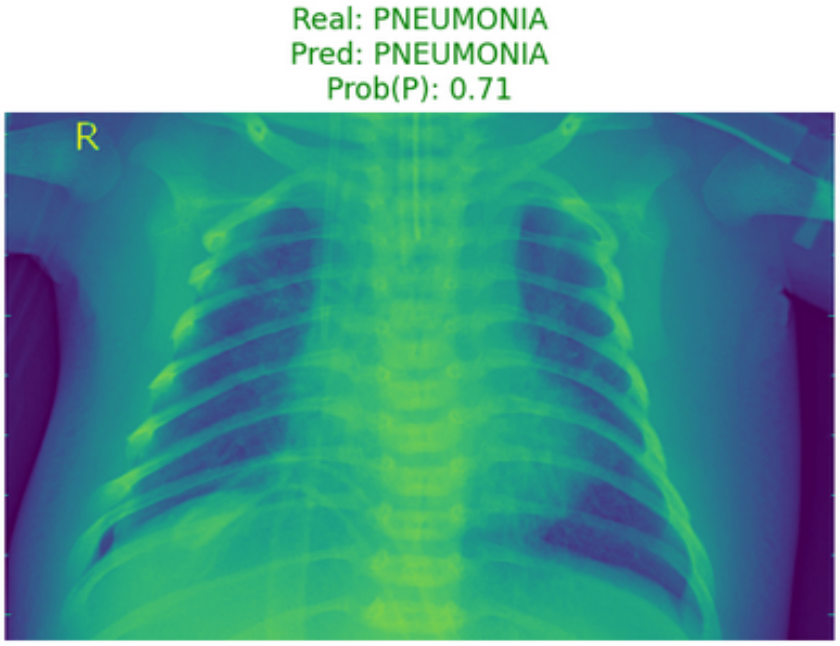
Informe de Desarrollo: Modelo de Detección de Neumonía por Radiografía



Proyecto para el Instituto Superior Politécnico de Córdoba
Procesamiento de imágenes y Modelos de IA.
Profesores: Sol Figueroa y Carlos Charletti.

1. Introducción y Objetivo del Proyecto

El objetivo principal de este proyecto fue desarrollar un modelo de Deep Learning capaz de clasificar imágenes de radiografías de tórax para identificar la presencia de neumonía. Se buscaba crear una herramienta de apoyo al diagnóstico que mejorará la eficiencia y precisión en la detección de esta condición. Para ello, se trabajó con un dataset de imágenes de radiografías de tórax (clases "NORMAL" y "PNEUMONIA"), divididas en conjuntos de entrenamiento, validación y prueba.

dataset: https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

2. Preparación y Aumento de Datos (Data Augmentation)

Desde el inicio del proyecto, se implementó una estrategia robusta de **aumento de datos** para mitigar el sobreajuste y mejorar la capacidad de generalización del modelo, dado el tamaño limitado del dataset. Todas las imágenes fueron reescaladas a **150x150 píxeles** y normalizadas a un rango de píxeles [0,1].

Las técnicas de aumento de datos aplicadas incluyeron:

- Reescalado: 1./255
- Rotación: rotation_range=20
- Desplazamiento: width_shift_range=0.2, height_shift_range=0.2
- Cizallamiento: shear_range=0.25
- Zoom: zoom_range=0.25
- Ajuste de Brillo: brightness_range=[0.7, 1.3]
- Ajuste de Canales: channel_shift_range=75
- Modo de Relleno: fill_mode='nearest'

3. Iteraciones de Modelado y Resultados Detallados

A lo largo del desarrollo, el principal desafío fue combatir un **sobreajuste severo y persistente**, manifestado por un **sesgo de predicción única** (el modelo tendía a predecir siempre una misma clase) en el conjunto de validación. A continuación, se detallan las distintas iteraciones de diseño y ajuste del modelo, junto con sus resultados:

Intento Nº1: CNN desde Cero - Configuración Inicial

Arquitectura: Una Red Neuronal Convolucional (CNN) construida desde cero. Se iniciaron las pruebas con una configuración base de capas convolucionales y de pooling.

Comportamiento Observado:

- Las etiquetas reales para las imágenes normales eran 0, pero las predicciones de probabilidad eran muy altas (~0.97 - 0.99).
- Al aplicar un umbral de 0.5, todas las predicciones resultaron en 1 (Neumonía).
- Pérdida en Test: 2.6550 (muy alta), indicando una nula capacidad de generalización.
- Precisión en Test: 0.5000, reflejando que el modelo predecía todo como "Neumonía" y, dado un dataset balanceado, solo acertaba el 50% de las veces.

Conclusión: El modelo no estaba aprendiendo a distinguir entre las clases; simplemente predecía la clase "Neumonía" de forma indiscriminada.

Intento Nº2: CNN desde Cero - Aumento de Capacidad (Más Filtros)

Acción Tomada: Se incrementó la capacidad de la red **añadiendo más filtros** a las capas convolucionales (ej. de 32-64-128 a **64-128-256**). La hipótesis era darle al modelo más "neuronas" y herramientas para aprender patrones más complejos.

Comportamiento Observado:

- loss en entrenamiento: Disminuía constantemente (de 2.9182 a 1.2375), lo que confirmaba que el modelo memorizaba los datos de entrenamiento.
- accuracy en entrenamiento: Aumentaba significativamente (de 0.7227 a 0.8403).
- val_accuracy (validación): Se estancó en 0.5052 (aproximadamente 0.50), sin mejora real.
- val_loss (validación): Se disparó enormemente (de 1.3703 a 24.3714), la señal más clara de sobreajuste.
- Evaluación en Test: loss: 2.3092, accuracy: 0.6218. Aunque la precisión en test mejoró ligeramente, la pérdida seguía siendo altísima, indicando una generalización deficiente.

Conclusión: El modelo seguía prediciendo una sola clase en validación/test, y el aumento de filtros solo facilitó que memorizara en lugar de generalizar.

Intento Nº3: CNN desde Cero - Reducción del Learning Rate y Adición de Early Stopping

Acción Tomada: Ante el sobreajuste severo, se redujo el learning rate de 0.0001 a un valor muy bajo (0.000005). La intención era permitir que el optimizador realizara pasos más pequeños y estables. También se añadió un callback EarlyStopping para monitorear la val_loss y detener el

entrenamiento cuando no hubiera mejoras significativas, restaurando los mejores pesos del modelo.

Comportamiento Observado:

- loss y val_loss: Ambos disminuyeron y se mantuvieron muy cerca (ej. 0.0970 vs 0.0943), lo cual era bueno en términos de evitar el sobreajuste.
- accuracy en entrenamiento: Muy baja y errática (de 0.4013 a 0.4727).
- val_accuracy: Se mantuvo estancada en 0.4948 (aproximadamente 0.50).
- Predicciones Específicas: Para imágenes NORMAL, el modelo predecía NORMAL (probabilidades ~0.3-0.4). Para imágenes PNEUMONIA, el modelo también predecía NORMAL (probabilidades ~0.3-0.4)!
- Evaluación en Test: loss: 0.0943, accuracy: 0.4979. Consistentemente alrededor del 50%.

Conclusión: El learning_rate excesivamente bajo impidió que el modelo aprendiera las características distintivas de la neumonía. La red se quedó "atrapada" en un estado donde siempre predecía "Normal".z

Intento Nº4: CNN desde Cero - Ajuste del Learning Rate (Ligero Aumento) y Regularización Reforzada

Acción Tomada: Se buscó un balance ajustando nuevamente el learning_rate de 0.00001 a 0.00005 (un ligero aumento respecto al intento anterior, pero aún muy bajo). Dropout: Aumentado a 0.6 (o 0.7). kernel_regularizer: Aumentado regularizers.l2(0.005) o 0.01 en las capas Dense.

Comportamiento Observado:

- Métricas de Entrenamiento: loss disminuye, accuracy aumentaba (ej. hasta 0.8514), precision, recall, auc mejoraban. Esto indicaba que el modelo memorizaba los datos de entrenamiento muy bien.
- val_loss (validación): Empieza en 0.9500 y luego aumenta significativamente (ej. a 1.4811).
- val_accuracy y val_precision (validación): Estancadas en 0.5042 - 0.5063.
- val_recall (validación): ¡Se mantuvo en 1.0000 (100%) en todas las épocas! Esto significaba que el modelo acertaba todas las neumonías reales, pero lo hacía clasificando *todas* las imágenes como "Neumonía", incluyendo las normales.
- Evaluación en Test: loss: 0.9538, accuracy: 0.5000, precision: 0.5000, recall: 1.0000. Consistentemente sesgado.

Conclusión: La CNN entrenada desde cero, incluso con aumento de datos y fuertes regularizaciones, no lograba aprender las características discriminatorias de forma generalizable.

Tenía demasiada capacidad para la cantidad de datos, lo que la llevaba a memorizar patrones en lugar de entenderlos.

Intento Nº5: Exploración de AveragePooling2D (Alternativa de Pooling)

Acción Tomada: Se consideró y exploró la posibilidad de reemplazar las capas MaxPooling2D con AveragePooling2D. La hipótesis era que AveragePooling2D podría ser más suave, retener más información espacial y, potencialmente, ayudar al modelo a capturar características más sutiles.

Conclusión Previa: Aunque podía ofrecer ventajas teóricas, el problema subyacente de la incapacidad de la CNN desde cero para generalizar con el dataset actual era demasiado grande para ser resuelto solo con un cambio en el tipo de pooling.

Intento Nº6: Últimos cambios (batch size)

- **Acción Tomada:** Se volvió a usar MaxPooling, se bajó los filtros de nuevo a 32, 64, 128 por que al parecer era lo adecuado para la cantidad de datos y para probar algo nuevo bajamos el batch size a 16 Para probar gradientes un poco más "ruidosos" (que a veces ayudan a la convergencia). Serían 480/16=30 actualizaciones de pesos por época.
- **Conclusión Previa:** La raíz del problema sigue siendo la capacidad del modelo en relación con la escasez de datos. Un batch_size más pequeño puede generar gradientes más ruidosos que *podrían* ayudar a escapar de mínimos locales, pero no puede compensar la falta de patrones generalizables en el dataset para una red de esta complejidad.Lo que está sucediendo es que el modelo está aprendiendo muy bien de los datos de entrenamiento(métricas de precision y recall son razonables y es alto el accuary), ¡pero no está generalizando en absoluto a los datos de validación en val loss y val accuary, de nuevo predice solo una clase.

Punto de Quiebre: Agotamiento de Estrategias con CNN desde Cero

El problema fue que el modelo tenía que aprender desde 0 a distinguir formas, reconocer patrones complejos con pocos datos, por lo que solo memorizo las de entrenamiento, teniendo un sobreajuste extremo. Las redes convolucionales están diseñadas para aprender jerarquías de características: las primeras capas aprenden bordes y esquinas, las capas intermedias aprenden texturas y partes de objetos, y las capas profundas aprenden objetos completos.

Aprender esta jerarquía completa desde cero requiere una cantidad masiva de datos (millones de imágenes), que es lo que se usa para entrenar modelos como VGG16.

VGG16 fue entrenado con el dataset de **ImageNet**, que contiene **millones de imágenes** (más de 14 millones) y miles de categorías (1000 clases)..

Implementación de Transfer Learning

Para abordar las limitaciones de un dataset de imágenes de tamaño limitado y mejorar significativamente el rendimiento, se implementó la técnica de **Transfer Learning** utilizando el modelo **VGG16** como base.

El funcionamiento se basa en aprovechar el vasto "conocimiento" que VGG16 ya posee. Este modelo fue previamente entrenado en **ImageNet**, una enciclopedia masiva de millones de fotos de todo tipo, aprendiendo a reconocer características visuales básicas como texturas, formas y, al combinarlas, a identificar objetos complejos.

La estrategia se dividió en dos fases clave:

Fase 1: Entrenamiento Inicial de la Cabeza Clasificatoria

Inicialmente, se cargó el modelo VGG16 pre-entrenado en ImageNet. El paso crucial fue congelar todas sus capas convolucionales (layer.trainable = False). Esto convirtió a VGG16 en un extractor de características fijas: su "cerebro experto" se utilizaría para analizar las radiografías sin modificar su conocimiento ya adquirido. Así, VGG16 procesaría una radiografía y devolvería una descripción detallada de los patrones, texturas y formas que veía.

Dado que VGG16 original está diseñado para clasificar 1000 tipos diferentes de imágenes (de ImageNet), y nosotros solo necesitamos clasificar dos (Normal o Neumonía), se le indicó a Keras no incluir el "head" original de VGG16 (include_top=False). En su lugar, se añadió una "cabeza" clasificatoria personalizada sobre el VGG16 congelado. Este "head" es un conjunto de capas nuevas, principalmente capas Dense y Dropout, conectadas directamente a la salida de las características extraídas por VGG16. En esta fase, solo los pesos de esta pequeña "cabeza" personalizada fueron entrenados para que aprendiera a interpretar las características de alto nivel generadas por VGG16 para la tarea específica de detección de neumonía.

- **Resultados al final de esta fase (en el conjunto de prueba):**
  - Accuracy: ~76.71%
  - Precision: ~95.62%
  - Recall: ~55.98%
  - AUC: ~0.9340

Aunque estos resultados iniciales eran prometedores en precisión, el Recall (capacidad de detectar casos de neumonía) mostraba un margen considerable de mejora, indicando que el modelo aún pasaba por alto una proporción significativa de casos reales.

Fase 2: Fine-Tuning (Ajuste Fino)

Para optimizar el rendimiento, especialmente el Recall, se procedió al **Fine-Tuning**. En esta etapa, en lugar de mantener todas las capas de VGG16 congeladas, se **descongelaron las últimas 4 capas convolucionales** (num_layers_to_unfreeze = 4) del modelo base (base_model.layers[-num_layers_to_unfreeze:]). Estas capas fueron marcadas como entrenables (layer.trainable = True).

Las **últimas capas** de una CNN pre-entrenada son cruciales porque, a diferencia de las primeras (que aprenden características muy generales como líneas y círculos), estas aprenden **características de más alto nivel y más específicas** (como patrones que podrían indicar neumonía). Al "descongelarlas", se le dio al modelo la oportunidad de que estas capas más profundas se **adaptaran y especializaran** aún más en los patrones únicos de las radiografías, en lugar de solo usar los patrones generales de ImageNet (que incluyen fotos de gatos, coches, etc.).

Esto permitió que tanto estas capas descongeladas de VGG16 como la cabeza clasificatoria ajustaran ligeramente sus pesos. El modelo se re-entrenó con una **tasa de aprendizaje (learning rate) muy baja** para preservar el conocimiento pre-existente de VGG16 y asegurar un ajuste fino y no una "re-escritura" completa del aprendizaje.

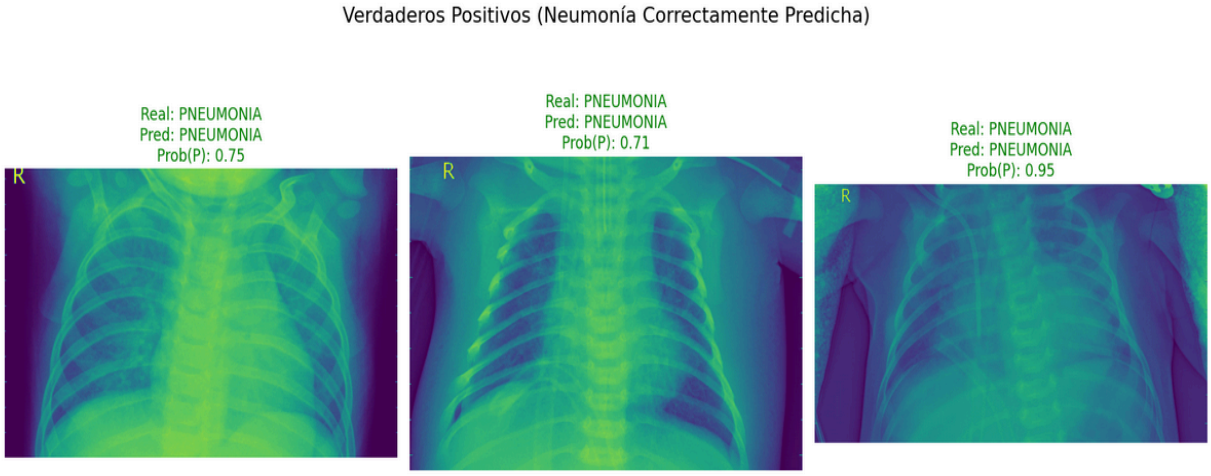
- **Resultados Finales del Modelo (Fine-Tuned) en el conjunto de prueba:**
    - Loss: 0.4427
    - Accuracy: 0.9103 (91.03%)
    - Precision: 0.9138 (91.38%)
    - Recall: 0.9060 (90.60%)
    - AUC: 0.9608

Conclusión: Los resultados finales del modelo con Fine-Tuning demuestran un **rendimiento excelente y equilibrado**. Con una precisión, recall y precisión específica por encima del 90%, el modelo es altamente efectivo para la detección de neumonía. Esto valida la potencia del Transfer Learning para tareas de clasificación de imágenes con datasets limitados y resalta la importancia del ajuste fino para maximizar el rendimiento.

Finalmente, la adopción de la estrategia de **Transfer Learning con VGG16** resultó ser la solución definitiva. El modelo resultante es robusto y ofrece un rendimiento prometedor, especialmente en términos de neumonía, lo que reduce las alarmas falsas.

Consideraciones Clínicas: Para una aplicación práctica, el modelo necesitaría una validación rigurosa por parte de expertos médicos y una calibración del umbral de decisión para adaptarse a necesidades clínicas específicas (ej., si se prefiere un mayor Recall a expensas de una ligera disminución en la Precision).

Conclusión final



El proceso iterativo de modelado ha confirmado que el sobreajuste fue el principal obstáculo en el desarrollo del modelo. Las reiteradas modificaciones en la arquitectura de la CNN desde cero (filtros, learning_rate, regularización, pooling) no lograron superar este problema fundamental de generalización.

Finalmente, la adopción de la estrategia de **Transfer Learning con VGG16** resultó ser la solución definitiva. El modelo resultante es robusto y ofrece un rendimiento prometedor, especialmente en términos de neumonía, lo que reduce las alarmas falsas.

Consideraciones Clínicas: Para una aplicación práctica, el modelo necesitaría una validación rigurosa por parte de expertos médicos y una calibración del umbral de decisión para adaptarse a necesidades clínicas específicas (ej., si se prefiere un mayor Recall a expensas de una ligera disminución en la Precision).