1. **Problema que resuelve:**

Consideramos que se requiere de conocimiento médico para lograr detectar de manera visual si existe o no una infección en los pulmones; sin embargo, en los RX normales es más abundante el color negro y se ve más libre de partículas extrañas, y en los RX con neumonía es más abundante el color blanco y se logran detectar partículas distribuidas por el pulmón, dicho esto podríamos tener una aproximación del diagnóstico, pero no es 100% confiable.

**RX:**

1. **Normal**: se logra detectar que en un RX normal el contorno del pulmón tiene forma definida, sin alteraciones. Se visualiza de forma más clara una especie de rama o red que inicia en el centro de la radiografía y se distribuye hacia ambos pulmones.
2. **Pneumonía**: En general las imágenes son más opacas y poco nítidas, los bordes de los pulmones se ven más irregulares.

**Identificación de una Pneumonía:**

Se clasifican en pneumonía viral y bacteriana. Los RX con pneumonía bacteriana tienen un color blanco opaco que sobresale y es homogéneo, se asimila a motas de algodón compactas; en los RX con pneumonía viral se detecta igualmente el color blanco que sobresale formando pequeños trozos desprendidos.



1. **Técnica de la solución:**

Las redes convolutivas ayudan a la solución del problema planteado, debido a que la Red Neuronal puede ser entrenada a partir de imágenes utilizando solo píxeles y etiquetas de enfermedades como entradas. Las redes neuronales convolutivas constan de múltiples capas diseñadas para requerir un preprocesamiento relativamente pequeño en comparación con otros algoritmos de clasificación de imágenes.

1. **Implementación:**
   1. **Definición del dataset:**

El conjunto de datos está organizado en 3 carpetas (entrenamiento, prueba, val) y contiene subcarpetas para cada categoría de imagen (neumonía / normal). Hay 5.863 imágenes de rayos X (JPEG) y 2 categorías (neumonía / normal).

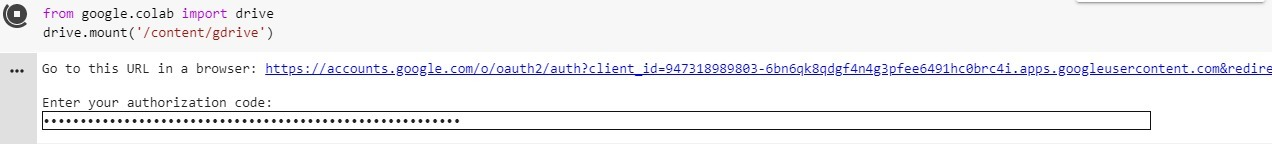
1. **Dataset test**: 234 imágenes normales y 390 con pneumonía.
2. **Dataset train**: 1342 imágenes normales y 3876 con pneumonía.
3. **Dateset val:** 8 imágenes normales y 8 con pneumonía.

Se seleccionaron imágenes de rayos X de tórax (anterior-posterior) de cohortes retrospectivas de pacientes pediátricos de uno a cinco años del Centro Médico para Mujeres y Niños de Guangzhou, Guangzhou. Todas las imágenes de rayos X del tórax se realizaron como parte de la atención clínica de rutina de los pacientes.

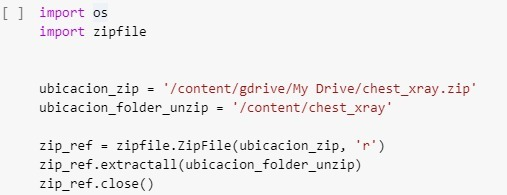
Para el análisis de imágenes de rayos X de tórax, todas las radiografías de tórax se examinaron inicialmente para el control de calidad eliminando todos los escaneos de baja calidad o ilegibles. Luego, dos médicos expertos calificaron los diagnósticos de las imágenes antes de autorizar el entrenamiento del sistema de IA. Para tener en cuenta los errores de calificación, el conjunto de evaluación también fue verificado por un tercer experto.

* 1. **Pasos:**

1. Carga de los dataset

Se carga las imágenes en la ruta <https://drive.google.com/file/d/16ajJdjHzPr_WwfauEOv5dKZ1IKYoMQyk/view?usp=sharing> Y se llama de la siguiente manera:

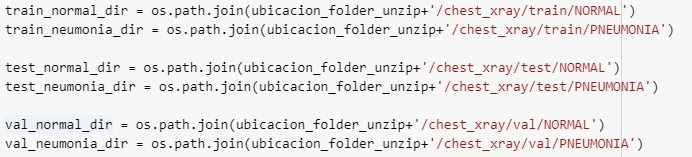
1. Importar y descomprimir dataset



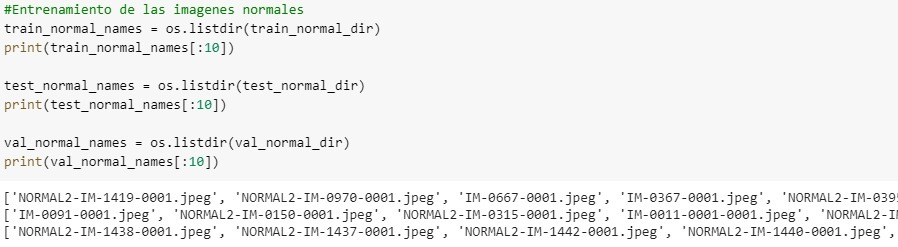
Así se ve las imágenes del dataset descomprimidas en Google Colaboratory el cual es usado para usar las virtudes del GPU:



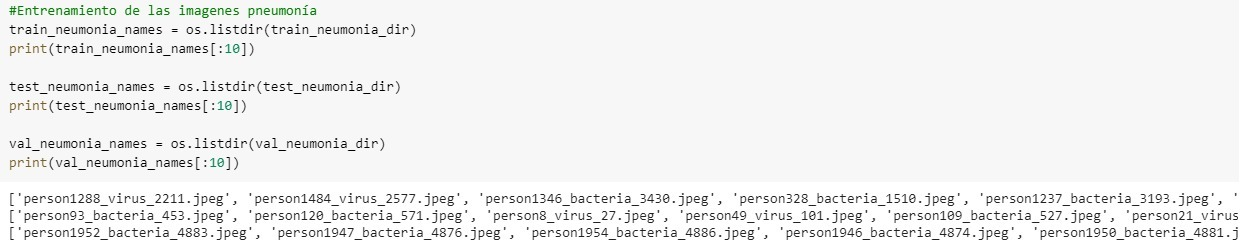
1. Se captura en variables los directorios del dataset según clasificación de las imágenes:



1. Entrenar las imágenes sin pneumonía



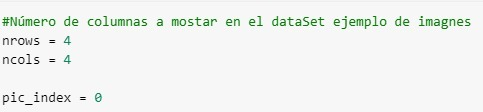
1. Entrenar las imágenes con pneumonía

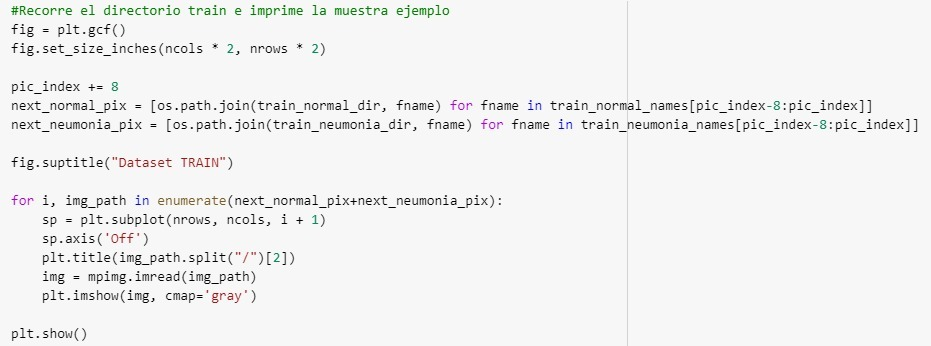


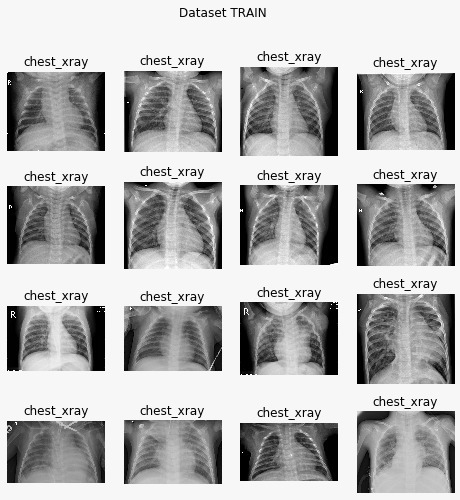
1. Importar numpy para el cálculo de matrices, tf para las redes neuronales, keras para el procesamiento de imágenes y matplotlib para el procesamiento de imágenes.



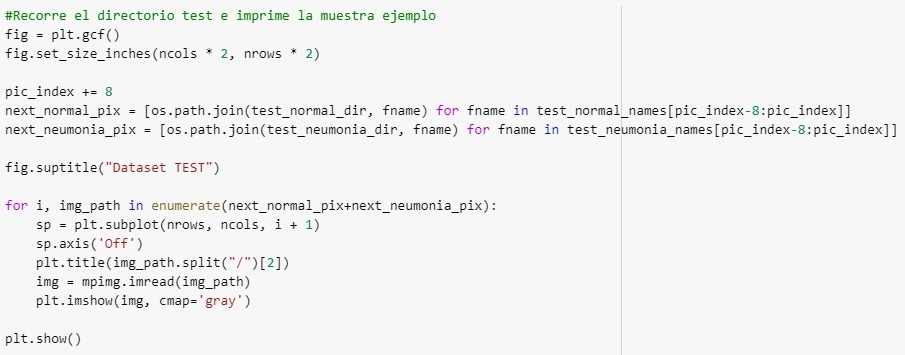
1. Ejemplo de las imágenes cargadas para el dataset Train

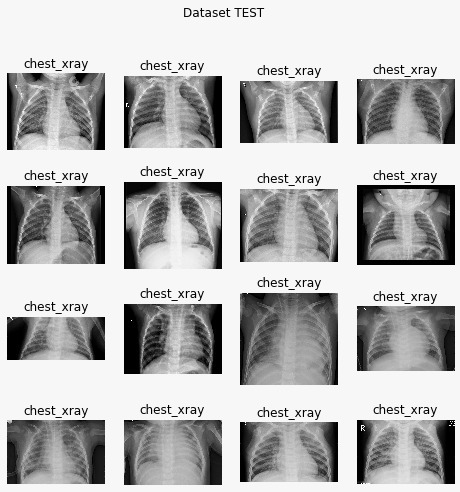




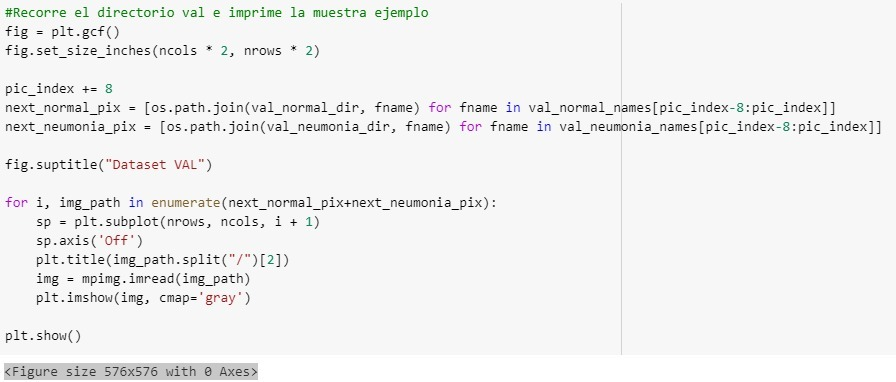


1. Ejemplo de las imágenes cargadas para el dataset Test

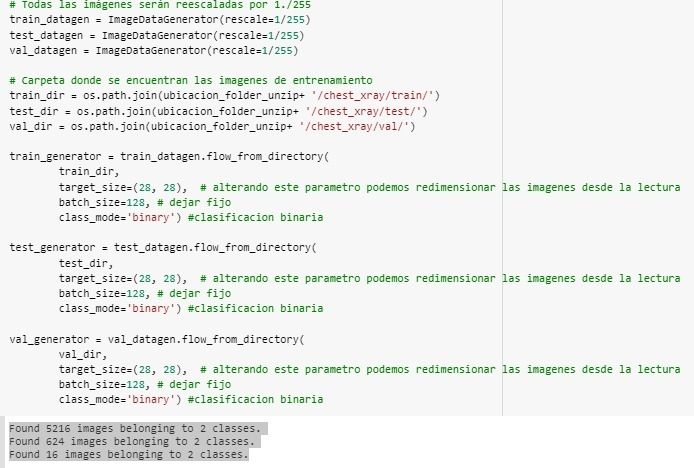




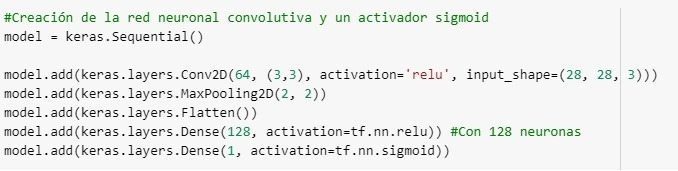
1. Ejemplo de las imágenes cargadas para el dataset Val



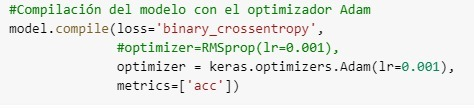
1. Preparación de imágenes previo al entrenamiento para cada dataset



1. Creación del modelo de red neuronal con TensorFlow

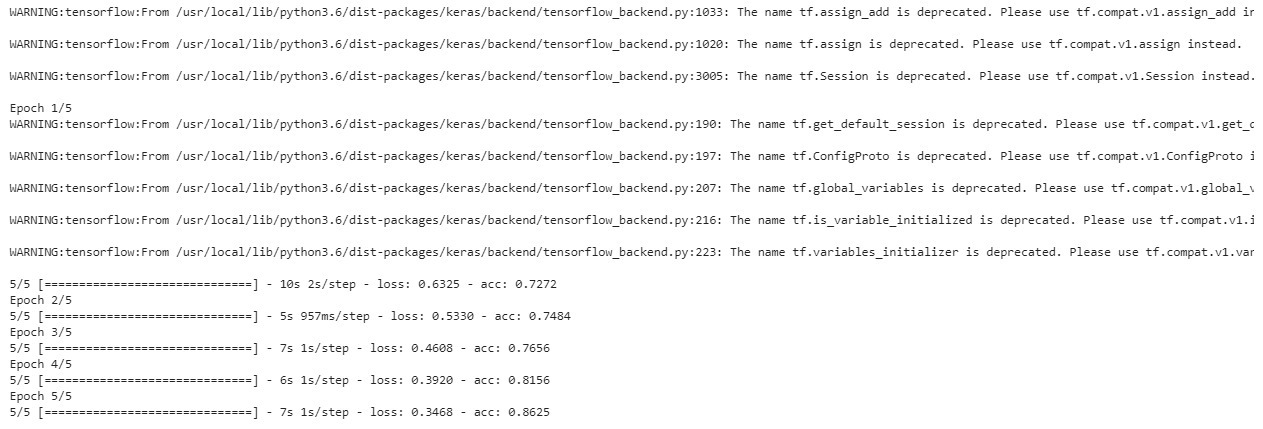


1. Compilación de modelo con parámetros de ajuste de dataset, rendimiento y calidad del modelo.

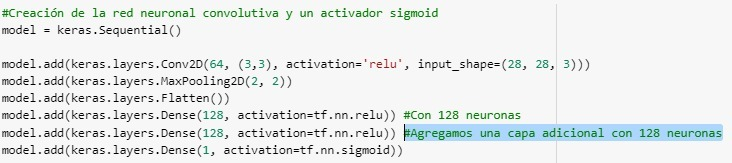


1. Entrenamiento con el dataset train

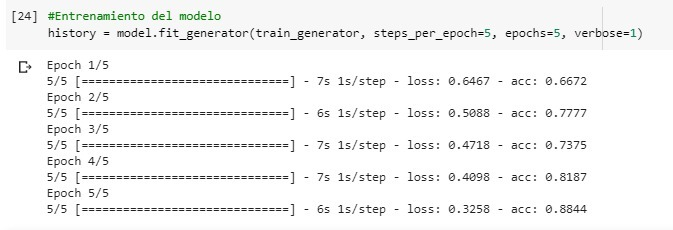




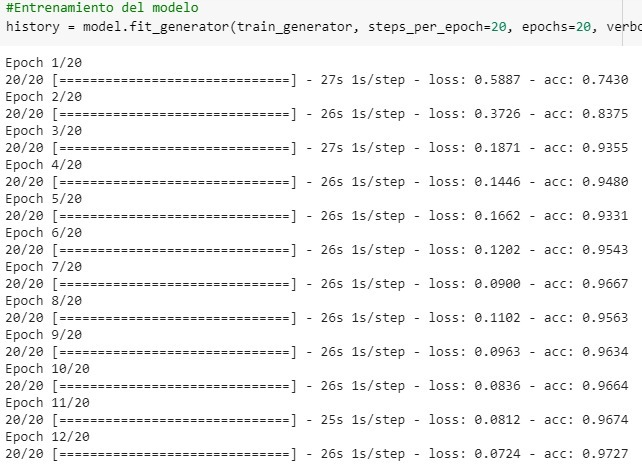
1. Añadiendo una capa más a la red neuronal del paso XI.

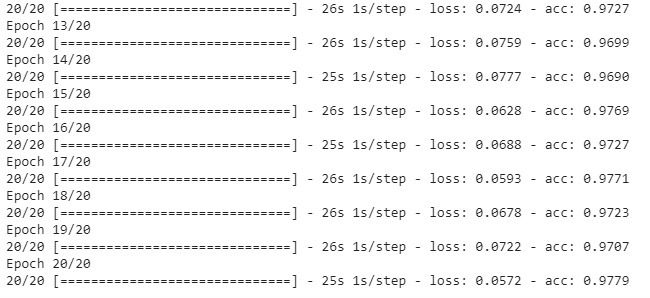


Reentrenamiento con la capa adicional de 128 neuronas.



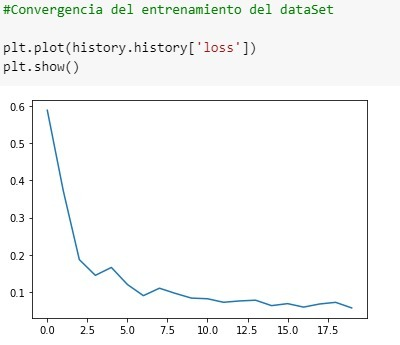
1. Añadiendo más pasos de las épocas, más épocas y reentrenando la red nuevamente.





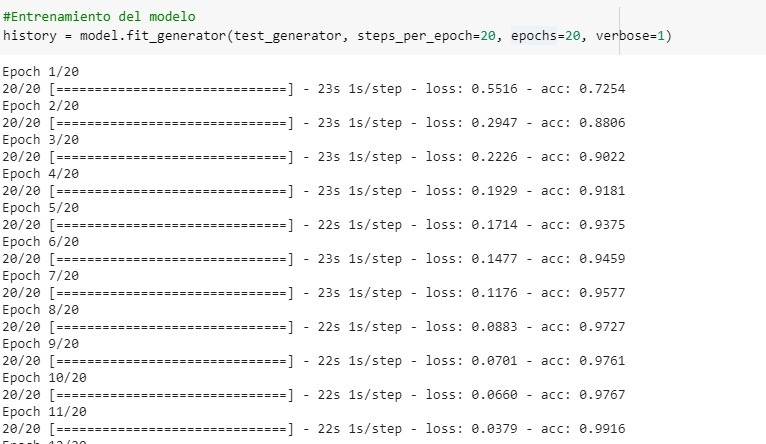
La calidad del modelo es aceptable (0.9779) con el dataset train entrenado con dos capas de 128 neuronas cada una, 20 épocas y 20 pasos por épocas.

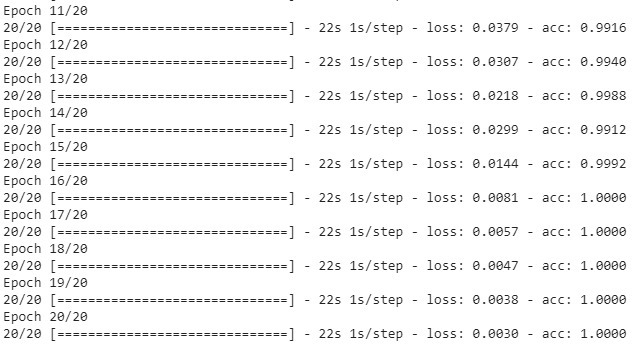
1. Gráfica de convergencia para dataset train



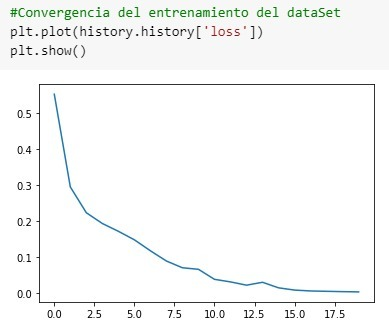
1. Reentrenando red neuronal con dataset test y las mismas condiciones anteriores.

Se recompila nuevamente el modelo para garantizar el reentrenamiento limpio.



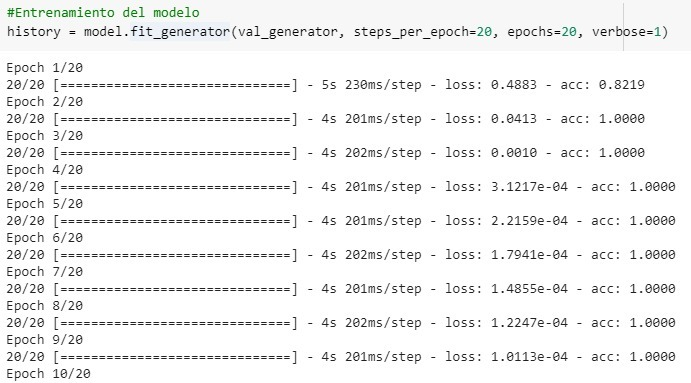


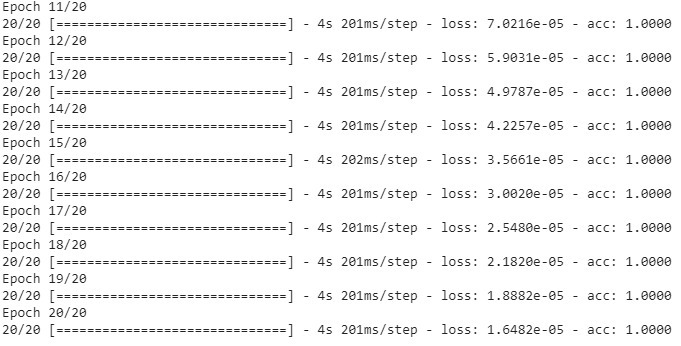
1. Gráfica de convergencia para dataset test



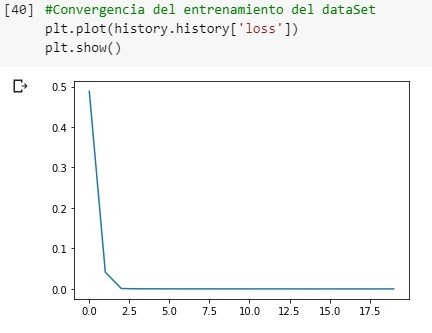
1. Reentrenando red neuronal con dataset val y las mismas condiciones anteriores.

Se recompila nuevamente el modelo para garantizar el reentrenamiento limpio.





1. Gráfica de convergencia para dataset val



**Conclusión:**

Del modelo neuronal indicado configurado con 2 capas de 128 neuronas, 20 épocas y 20 pasos por época, se puede decir que la convergencia es lenta y el modelo es confiable para dataset grandes como train, como se evidencia en el paso XV en comparación con el entramiento de la red con test (Paso XVII) y con val (Paso XIX).

\*Que la calidad de modelo (acc) sea 1.00 significa que el entrenamiento no es confiable, pues de ser así significaría que la probabilidad de acierto de la predicción con una imagen adicional es siempre 100%, lo cual es absurdo.

**Código de implementación**

<https://colab.research.google.com/drive/13J4MjuV5cc-TpUZO2JNu86sYknnaKvUL>

**Referencias:**

<https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>

<https://www.marktechpost.com/2019/10/16/classify-handwritten-digits-with-tensorflow/>