Detección de COVID-19 en imágenes de rayos X mediante el uso de Redes Neuronales de Convolución

Sherlyn Ballestero Cruz SHERLYNBALLESTERO@GMAIL.COM
Grupo C113

Rolando Sánchez Ramos ROLSANCHEZ@YANDEX.COM
Grupo C211

Tutor(es):
MSc. Alejandro Piad Morphis, *Universidad de La Habana*

Resumen

El objetivo principal de este trabajo es investigar y comparar varias técnicas de Deep Learning aplicadas a imágenes de rayos X de tórax sobre tres clases (pacientes infectados con COVID-19, pacientes con neumonía y pacientes sanos) para la clasificación de las mismas de acuerdo al tipo que representen. En este proceso se utilizaron Redes Neuronales de Convolución, las cuales posibilitan la creación de manera eficiente de modelos de aprendizaje automático supervisado orientados a tareas de clasificación a partir de una serie de imágenes previamente etiquetadas. Específicamente se utilizaron los modelos de redes preentrenadas DenseNet201, EfficientNetB7 y MobileNet, como parte de la técnica de Transfer Learning, las cuales fueron ejecutadas a partir de las implementaciones que ofrece la API de Keras para el lenguaje Python. Finalmente, se mezclaron los tres modelos mencionados mediante el uso de Ensemble Learning como método para mejorar la precisión final.

Abstract

The main purpose of this work is to investigate and compare several Deep Learning techniques applied to ray X chest images from three classes (patiens infected with COVID-19, patiens with pneumonia and normal patiens) in order to classify them according to their type. For this process we used Convolutional Neural Networks, wich allow us to create efficient supervised machine learning models that can be oriented to classification tasks given previously tagged images. In this case, we used the pretrained networks DenseNet201, EfficientNetB7 and MobileNet, as part of Transfer Learning technique, wich were executed by using the implementations that Keras API for Python language offers us. Finally, we merged the three models mentioned before by using Ensemble Learning as a method for improving the final accuracy.

Palabras Clave: Red Neuronal de Convolución, Ensemble Learning, Red Preentrenada, Transfer Learning.
Tema: Matemática aplicada, Deep Learning, Deep Computer Vision.

1. Introducción

La COVID-19 es una enfermedad infecciosa causada por el SARS-CoV-2, que tuvo un primer brote en la ciudad de Wuhan en diciembre de 2019 y para enero del 2020 ya se habían comunicado los primeros casos en Tailandia y Japón. Cuando menos lo esperábamos ya había alcanzado la mayor parte del mundo. Las personas con sospecha de COVID-19 deben saber lo antes posible si están infectadas, para poder aislarse, recibir tratamiento y comunicarlo a sus contactos cercanos. Actualmente se requiere para el diagnóstico formal una prueba de laboratorio (RT-PCR) que necesita un equipo especializado y tarda al menos 24 horas en producir resultados. Los médicos pueden utilizar imágenes de tórax para diagnosticar a las personas con síntomas mientras se espera los resultados del RT-PCR, o cuando los resultados son negativos y la persona se encuentra con síntomas. Un inconveniente es que se necesita de un experto en radiología, lo cual requiere un tiempo considerable. Por lo tanto, es necesario desarrollar un sistema de análisis automatizado para ahorrar a los profesionales médicos un tiempo valioso. Para la solución de este problema proponemos la utilización de una forma automatizada de análisis de radiografías a través de Redes Neuronales de Convolución[1] ya preentrenadas, basándonos en la técnica de Transfer Learning y el uso del método de Ensemble Learning. Con el objetivo de llevar a cabo de forma experimental esta tarea, se utilizarán las implementaciones de dichas herramientas en la API de Keras para el lenguaje de programación Python y el IDE Jupyter Notebook, a partir de las cuales podremos ejecutar los modelos mencionados anteriormente y evaluar sus resultados mediante el uso de métricas de acierto, el visualizado del mejoramiento de estas durante la fase de entrenamiento y la detección de las regiones más influyentes en las imágenes analizadas para su clasificación.

2. Obtención del Conjunto de Datos

El conjunto de imágenes para la realización de este proyecto fue conformado principalmente a partir de dos repositorios públicos disponibles de tomografías de rayos X de tórax anónimas. El primero de estos fue un conjunto de imágenes de pacientes con COVID-19, elaborado por el Dr. Joseph Cohen, becario postdoctoral en la Universidad de Montreal. Desde marzo del 2020, el Dr. Cohen comenzó a recopilar imágenes de rayos X de los casos de COVID-19 y a publicarlas en su repositorio de GitHub[2], del cual se obtuvieron un total de 243 imágenes en formato png. El segundo de los repositorios analizados fue un conjunto de imágenes de pacientes con neumonía y pacientes sanos publicado en Kaggle[3], a partir del cual se obtuvieron un total de 188 imágenes en formato jpg, de las cuales una mitad correspondía a pacientes con neumonía y la otra a pacientes sanos.

3. Visualización y Preprocesamiento del Conjunto de Datos

El primer paso para organizar el conjunto de datos fue dividir el mismo en los conjuntos *train* y *test*, los cuales representan el conjunto de datos de entrenamiento y el conjunto de datos de comprobación respectivamente. El conjunto *train*, tal y como su nombre sugiere, contiene los ejemplos de imágenes que le serán dados a los modelos de redes neuronales propuestos para su entrenamiento y ajuste de hiperparámetros, luego el conjunto *test* deberá contener ejemplos que permitirán evaluar la calidad de las predicciones una vez entrenados los modelos utilizados.

Para la conformación del conjunto *train*, se escogieron en total 248 imágenes divididas entre las tres clases a predecir. En este caso se utilizaron 100 imágenes correspondientes a pacientes con COVID-19, y 74 imágenes representativas tanto para pacientes con neumonía como para aquellos etiquetados como saludables.

De forma análoga a la elaboración del conjunto *train*, el conjunto *test* fue confeccionado por 90 imágenes en total, de las cuales 50 representan a convalecientes de COVID-19, mientras que se seleccionaron 20 tanto para pacientes enfermos de neumonía como para pacientes sanos.

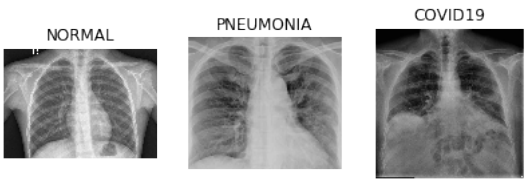

Figura 1: Ejemplos de imágenes de las clases COVID19, Neumonía y Normal.

Posteriormente se reescalaron todas las imágenes de cada uno de los conjuntos de datos a utilizar, específicamente se empleó la clase *ImageDataGenerator*(perteneciente a la biblioteca Keras) con la cual se logró reescalar las imágenes a un tamaño de $224x224$ píxeles, se normalizaron en el rango $[0; 1]$, se preprocesaron las mismas para dividirlas en canales RGB y finalmente se aplicaron un zoom y rotación leves sobre estas con el fin de mejorar su calidad al ser analizadas.

4. Transfer Learning

El Transfer Learning[4] se muestra como una de las técnicas más importantes del Deep Learning para el aprendizaje automático en inteligencia artificial. Consiste en esencia, en aprovechar una gran cantidad de información relacionada con la solución de un problema y utilizarla sobre otro distinto con el que comparta ciertas características, o sea modificar patrones ya entrenados(o redes neuronales) para reconocer ciertas características. Si se desea construir un clasificador de imágenes pero no se cuenta con suficientes datos de entrenamiento, entonces es una buena idea utilizar las capas bajas de un modelo preentrenado. Específicamente en este caso utilizamos las redes preentrenadas: DenseNet201, MobileNet y EfficentNetB7, a las que se les agregan algunas capas para procesar la salida y adaptar la arquitectura a nuestro problema. La selección de los modelos mecionados se debe a que una estrategia para lidiar con redes neuronales es usar redes previamente preentrenadas con bases de datos grandes y adaptarlas al problema de nuestro interés, para ello es necesario que esta haya sido entrenada para resolver problemas con carácter más general, del que el nuestro se pueda considerar un caso particular. Las redes neuronales de convolución elegidas se utilizan en su versión preentrenada con el conjunto de datos procedente del ImageNet Challenge[5], el cual presenta alrededor de 1.4 millones de imágenes etiquetadas en más 1000 clases distintas.

4.1 DenseNet201

DenseNet201[6] es una red neuronal de convolución que está conformada por 201 capas de profundidad. Esta red posee tamaño de entrada de $224x224$ píxeles. En DenseNet201 cada capa obtiene entradas adicionales de todas sus capas precedentes y pasa su propio mapa de características a las capas subsecuentes. La idea de este modelo es que cada capa recibe un "conocimiento colectivo" de todas las capas que le anteceden. Debido a lo anteriormente mencionado, la red puede ser más fina y compacta, por lo tanto se obtiene una mayor eficiencia computacional y de memoria. Además, gracias a su arquitectura permite realizar tareas de clasificación estableciendo límites de decisión bastante flexibles, lo cual explica porqué DenseNet201 funciona tan bien cuando los datos de entrenamiento son insuficientes.

4.2 EfficientNetB7

EfficientNetB7[7] proviene de una familia de modelos llamada EfficientNet, las cuales son unas de las implementaciones más eficientes que existen en el campo de

las redes neuronales de convolución. Específicamente esta red posee 813 capas y recibe imágenes con dimensiones de $600x600$ píxeles. En el momento de su creación, este modelo era 8 veces más pequeño y 6 veces más rápido que la mejor red neuronal convolucional conocida en aquel entonces, por lo que se convirtió en una de las principales opciones para realizar tareas de clasificación mediante Transfer Learning con un orden de magnitud de parámetros bajo.

4.3 MobileNet

El modelo MobileNet[8], tal y como su nombre indica, está diseñado para usarse en aplicaciones móviles y es el primer modelo de visión por computadora móvil del framework para ciencia de datos TensorFlow[9].

MobileNet utiliza convoluciones separables en profundidad, lo que reduce significativamente el número de parámetros en comparación con una red de convoluciones regulares con la misma profundidad. Esto da como resultado redes neuronales profundas y ligeras.

Mobile Net es una clase de red neuronal de convolución creada por Google como proyecto de código abierto y, por lo tanto, esto nos brinda un excelente punto de partida para la creación de clasificadores increíblemente pequeños y rápidos.

5. Ensemble Learning

Ensemble Learning[10] es una técnica de aprendizaje automático que combina varios modelos base para producir un modelo predictivo óptimo. Es decir en lugar de hacer un modelo y esperar que este modelo sea el mejor(el más preciso predictor) que podemos hacer, esta táctica toma en cuenta una gran cantidad de modelos y los promedia para producir un modelo final.

En el caso específico de este trabajo se creó un modelo de Ensemble Learning mediante el método *concatenate*, perteneciente a la API de Keras, y con el cual pudimos mezclar los modelos mencionados anteriormente en uno solo y de esta manera conformar un clasificador por votación, el cual se puede entender como un modelo que para una imagen de entrada dada, brinda como salida la clase que mayor presencia tuvo en la clasificación interna individual de cada modelo contenido, lo cual en principio producirá resultados con mayor fiabilidad y seguridad.

6. Entrenamiento y Análisis de los Resultados

Para la realización del entrenamiento de los modelos se estableció como forma de parada el hecho de que alguna de las métricas tenidas en cuenta dejara de mejorar, en este caso la precisión de predicción y el valor de pérdida de la función de costo de cada una de las redes utilizadas.

Es importante mencionar que para la visualización de los resultados se implementaron algunos métodos muy útiles. Por ejemplo para mostrar las curvas de aprendizaje se implementó el método *display_learning_curves*, el cual sirvió como herramienta para mostrar la variación de las métricas antes mencionadas respecto al número de episodios de entrenamiento de los modelos. Además se implementó el método *make_gradcam_heatmap*[11], con el cual se lograron mostrar las zonas más influyentes en el proceso de clasificación de algunas imágenes.

Para cada uno de los modelos se utilizó como función de optimización Adam[12], con un ratio de aprendizaje de 0.0001.

6.1 Entrenamiento del Modelo DenseNet201

Para el entrenamiento de esta red neuronal se añadieron las capas *AveragePooling2D*, *Flatten*, *Dense* y *Dropout* a la salida de esta, permitiendo adaptar el modelo preentrenado a las condiciones del problema en cuestión. Luego, se establecieron 16 episodios de ejecución y en cada uno se procesaron lotes de 16 imágenes. Finalmente se obtuvo una precisión final del modelo de 96 % y 93 % para los conjuntos de entrenamiento y de prueba respectivamente, con lo cual se deduce que no se produjo overfitting, es decir, el modelo generaliza con gran precisión para conjuntos de datos que no había visto con anterioridad en su entrenamiento. Además, se obtuvieron los valores 0.0213 y 0.0262 como valores de pérdida de precisión del modelo de acuerdo a la función de pérdida utilizada, que fue en este caso Minimum Mean Square Error[13]. Por útlimo, es importante mencionar que el ratio de aprendizaje terminó con un valor de $5.0000e-05$.

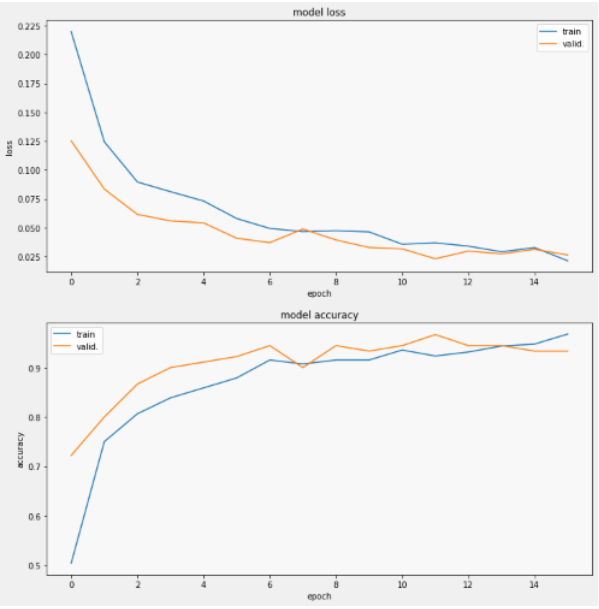

Figura 2: Gráfico de evolución de la pérdida y precisión de DenseNet201.

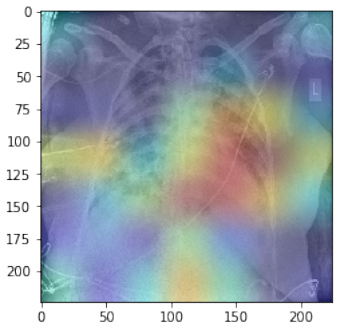

Figura 3: Ejemplo de zonas más influyentes en la predicción de DenseNet201 en un caso específico de COVID-19.

6.2 Entrenamiento del Modelo EfficientNetB7

En el caso de EfficientNetB7, se agregaron a la salida de esta las mismas capas que en el modelo DenseNet201. Además, se ejecutaron 9 episodios de entrenamiento de 15 posibles debido a que las métricas analizadas dejaron de mejorar, y se analizaron lotes de imágenes del mismo tamaño que en la red anterior.
Las precisiones finales obtenidas para los conjuntos de entrenamiento y de prueba fueron de 72 % y 75 % respectivamente. Se obtuvieron valores de pérdida de 0.1244 y 0.1216 de igual manera para los conjuntos anteriores utilizando la misma función de pérdida que en la red neuronal anterior, y el ratio de aprendizaje finalizó con un valor de $5.0000e-05$.

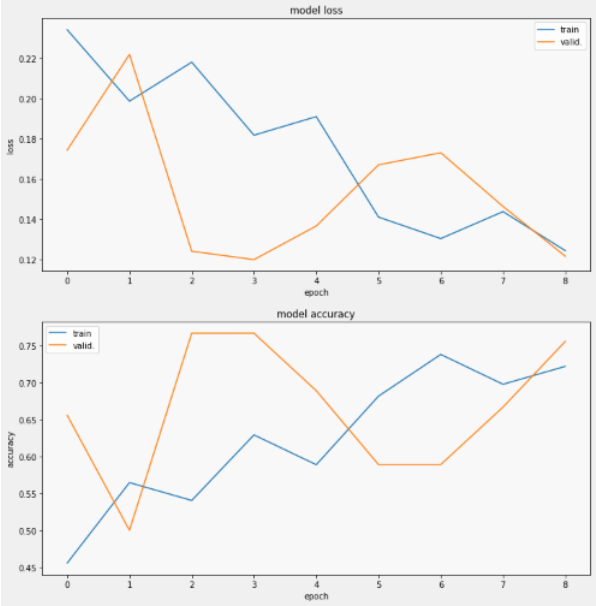

Figura 4: Gráfico de evolución de la pérdida y precisión de EfficientNetB7.

6.3 Entrenamiento del Modelo MobileNet

Para el entrenamiento de MobileNet se agregaron las mismas capas finales que en los modelos anteriores. Luego, se ejecutaron 16 episodios completos de entrenamiento donde en cada uno se analizaban conjuntos de 16 imágenes de ejemplo.
Una vez completada la ejecución de este modelo, se obtuvieron precisiones de 95 % y 94 % para los conjuntos de entrenamiento y de prueba respectivamente. Los valores de la función de pérdida fueron 0.4519 y 0.4177 respecto a dichos conjuntos y estos fueron calculados mediante la función de coste Categorical Smooth Loss[14]. El ratio de aprendizaje finalizó con el mismo valor que en los entrenamientos de los modelos anteriores.

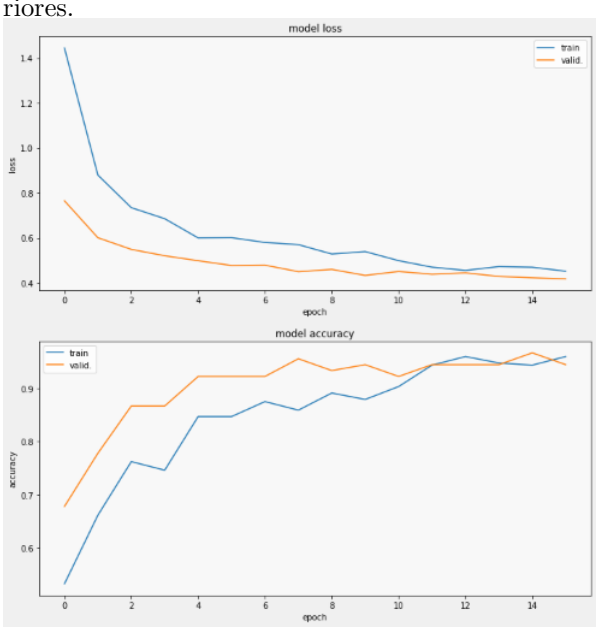

Figura 5: Gráfico de evolución de la pérdida y precisión de MobileNet.

6.4 Entrenamiento del Modelo de Ensemble Learning(DenseNet201, EfficientNetB7 y MobileNet)

Para conformar este modelo se concatenaron las salidas de los modelos DenseNet201, EfficientNetB7 y MobileNet en una sola y a esta se le agregó una capa *Dense*. Esta red se entrenó en 16 episodios con lotes de 16 imágenes cada uno. Luego se estableció como función de coste a Categorical Crossentropy[15]. Finalmente se obtuvieron predicciones de 57 % y 48 % para los conjuntos de datos de entrenamiento y de prueba respectivamente, por lo que se concluye que este modelo produce underfitting, lo que significa que el modelo no se ajustó lo suficiente al conjunto de datos de entrenamiento y por consiguiente, realiza una mala generalización de casos de prueba no vistos anteriormente.

7. Conclusiones

Con la realización de este trabajo se demuestra la utilidad, eficiencia y precisión de las redes neuronales convolucionales para la realización de tareas de clasificación de imágenes con un número limitado de recursos, lo cual las convierte en fuertes candidatas como herramientas a utilizar en los procesos de diagnóstico médico y detección de anomalías. Además, cabe destacar que los mejores resultados de predicción fueron de los modelos *DenseNet*201 y *MobileNet*, los cuales ejecutaron el proceso de entrenamiento en menor tiempo que

las otras redes neuronales y contituyen detectores de COVID-19 a partir de imágenes de rayos X de tórax con capacidad de ser puestos en producción y propuestos para ser utilizados directamente en los hospitales y centros médicos donde se traten a pacientes infectados con el coronavirus. Resulta a la vez interesante, la imagen obtenida de las zonas más influyentes en la predicción del modelo DenseNet201 para un caso de prueba de un paciente con COVID-19, lo cual consideramos sería vital para la creación de nuevos paradigmas de análisis de tomografías en la detección de afecciones. También, se hace relevante recalcar el potencial de las técnicas de Transfer Learning para optimizar el tiempo de implementación de redes neuronales eficaces en disímiles tareas de visión por computadora para los cuales se cuenta con un número muy bajo de datos en los conjuntos de entrenamiento y prueba. Finalmente, creemos que la técnica de Ensemble Learning, si bien no produjo los resultados esperados, es aconsejable la continuidad de su estudio y perfeccionamiento, lo cual llevará a la creación de modelos cada vez más confiables, basados en la predicción de otros múltiples modelos destinados a un mismo objetivo de clasificación.

8. Recomendaciones

Para la obtención de mejores resultados en el entrenamiento de modelos de aprendizaje automático en general se aconseja obtener la mayor cantidad de datos de calidad posibles, pues de esta forma los algoritmos logran ofrecer predicciones más cercanas a la realidad y con una precisión considerablemente mayor. Por otra parte, sería ideal no solo usar los conjuntos de datos de entrenamiento y de prueba, sino también la creación de un conjunto de validación, el cual permitirá determinar en una fase temprana si la calidad de las predicciones obtenidas son suficientemente buenas como para poner en evaluación el modelo con el conjunto de pruebas, por lo que se lograría detectar si el mismo está produciendo underfitting u overfitting con inmediatez. Finalmente, es importante realizar un estudio ulterior del uso de Ensemble Learning y el inmenso número de posibilidades que brinda para la construcción de modelos predictivos más robustos y seguros tanto para el diagnóstico de enfermedades a partir de tomografías de rayos X, como para tareas de detección de objetos como parte del campo de la visión por computadora.

Referencias

[1] Wikipedia. URL: http://es.m.wikipedia.org/Red_neuronal_convolucional. Consultado el 14 de noviembre de 2021.

[2] GitHub. URL: https://github.com/ieee8023/covid-chestxray-dataset. Consultado el 29 de octubre de 2021.

[3] Kaggle. URL: http://www.kaggle.com/khoongweihao/covid19-xray-dataset-train-test-sets. Consultado el 29 de octubre de 2021.

[4] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras and Tensorflow, Second Edition.* página 345.

[5] ImageNet. URL: http://www.image-net.org. Consultado el 14 de noviembre de 2021.

[6] ArVix. URL: https://arxiv.org/abs/1608.06993. Consultado el 14 de noviembre de 2021.

[7] ArVix. URL: https://arxiv.org/abs/1905.11946. Consultado el 14 de noviembre de 2021.

[8] PrabinNepal. URL: https://prabinnepal.com/mobilenet-architecture-explained/. Consultado el 14 de noviembre de 2021.

[9] TensorFlow. URL: https://www.tensorflow.org. Consultado el 14 de noviembre de 2021.

[10] Wikipedia. URL: https://en.m.wikipedia.org/wiki/Ensemble_le Consultado el 14 de noviembre de 2021.

[11] ArVix. URL: https://arxiv.org/abs/1610.02391. Consultado el 14 de noviembre de 2021.

[12] TopBigData. URL: https://topbigdata.es/optimizacion-de-gradiente-de-code-adan-desde-cero/. Consultado el 14 de noviembre de 2021.

[13] Wikipedia. URL: https://en.m.wikipedia.org/wiki/Minimum_m Consultado el 14 de noviembre de 2021.

[14] LinkedIn. URL: https://www.linkedin.com/pulse/label-smoothing-solving-overfitting-overconfidence-code-sobh-phd/. Consultado el 14 de noviembre de 2021.

[15] Medium. URL: https://link.medium.com/vslglB1Sdlb. Consultado el 14 de noviembre de 2021.