

Analítica avanzada

Evaluación de lesiones cutáneas usando redes neuronales (CNN)

01/10/2020 al 22/10/2020

Proyecto Final

Mónica Villasuso López

THE BRIDGE - Data Science Bootcamp

Visión General

Como reto final de la segunda mitad del bootcamp en Data Science, se requiere llevar a cabo un proyecto individual bajo las siguientes premisas:

- Una idea en la que se pudieran consolidar los conocimientos adquiridos durante la segunda parte del bootcamp (Analítica Avanzada - Machine Learning y Deep Learning).
- La evaluación de alternativas y la disponibilidad de datos o información para llevarlas a cabo,
- Un proyecto que además de académico generara un interés particular, con lo cual la motivación para encararlo sería mayor.

Tomando en cuenta lo anterior, y sumado al gran valor que agrega en el sector de la salud decidí orientar mi búsqueda hacia esta vía. Personalmente, siempre he tenido la preocupación por el daño que las radiaciones solares (y más aún en los últimos años) pueden generar en la piel. Es una realidad que en muchos casos este daño puede ser mortal (de hecho, tengo dos referencias cercanas que han fallecido por esta razón) pero es también una realidad que este daño se puede minimizar, y mucho, si se hace un diagnóstico temprano.

Por otro lado, son muchas las fuentes y publicaciones que hacen referencia a la revisión personal de las lesiones cutáneas siguiendo una regla básica. La “**Regla del ABCDE**” es recomendada para buscar algunos de los signos comunes de melanoma, que es una de las formas más mortales del cáncer de piel. Según esta regla, ante las lesiones en la piel (lunares, manchas) es necesario observar:

A - Asimetría

Una parte de un lunar o mancha de nacimiento (angioma) no es igual a la otra.

B - Borde

Los bordes son irregulares, desiguales, con rebordes, o borrosos.

C - Color

El color no es el mismo en todos lados y puede incluir matices de marrón o negro, a veces con zonas de color rosado, rojo, blanco o azul.

D - Diámetro

El lunar es más grande de ¼ de pulgada (0.6 cm) de ancho (aproximadamente el tamaño del borrador de un lápiz) aunque en ocasiones los melanomas podrían ser menores a este tamaño.

E - Evolución:

El lunar está cambiando en cuanto a tamaño, forma o color.

Debido a que muchas de estas señales son visibles y detectables a través de una imagen, decidí intentar validar si, con un modelo de aprendizaje supervisado y entrenando una Red Neuronal Convolutiva se podría llegar a una predicción. El resultado de la predicción tendría que ver con la posibilidad de que la lesión fuese **Sospechosa** (en cuyo caso se recomienda la revisión por un especialista) o **Normal**.

Es importante destacar el objeto lectivo o académico de este ejercicio pues el mismo no pretende ser una herramienta de diagnóstico. Para ello se requiere un análisis más exhaustivo,

datos más depurados y especializados y la validación o acompañamiento de un profesional en el área de la salud.

Objetivo

De acuerdo a los criterios de evaluación establecidos, he optado por el entregable A.

Especificaciones

Software

Python v3.8 o superior

Hardware

Procesador i5

Memoria RAM 8 GB

Espacio en disco 150 MB

Requerimientos

- decorator==4.4.2
- Flask==1.1.2
- jupyter==1.0.0
- jupyter-client==6.1.5
- jupyter-console==6.1.0
- jupyter-core==4.6.3
- matplotlib==3.2.2
- notebook==6.0.3
- numpy==1.19.0
- pandas==1.0.5
- python==3.8.3
- plotly==4.9.0
- scikit-learn==0.23.2
- scipy==1.5.1
- seaborn==0.10.1

• Contexto (Alcance y Planificación)

Las fases definidas para el proyecto, en función al alcance y los requerimientos académicos a cubrir son:

- 1) Alcance y Planificación del proyecto
- 2) Recolección y preparación de datos
- 3) Entrenamiento de varios modelos
- 4) Selección del mejor modelo y entonación de parámetros
- 5) Pruebas finales y conclusiones

• Recolección de datos

Debido a que se trata de un caso de Aprendizaje supervisado, se requiere que los datos se encuentren bien diferenciados. Para ello se obtuvieron imágenes de lesiones de la piel en diversos sitios y publicaciones en internet.

Las imágenes obtenidas, que alimentarán el modelo posteriormente se clasificaron según :

0 - **Malignos**: Fotografías de lunares, manchas o lesiones de la piel que luego de ser estudiadas han sido diagnosticadas como Malignas (Sospechosas) , es decir, con algún tipo de melanoma o condición que amerite tratamiento con un especialista

1 - **Benignos**: Fotografías de lunares, manchas o lesiones de la piel que luego de ser estudiadas han sido diagnosticadas como Benignas (Normales) sin que ameriten especial revisión médica.

En total se recolectaron para el análisis preliminar, 303 imágenes (a color y en diferentes tamaños y formatos). De éstas, 142 corresponden a lesiones Malignas y 161 a lesiones de naturaleza Benigna (Clases 0 y 1 respectivamente).

Siendo así el modelo a desarrollar sería un modelo de clasificación de imágenes en dos categorías siendo

- 1 (TRUE) cuando el resultado indique que la lesión es NORMAL y
- 0 (FALSE) cuando la lesión es SOSPECHOSA y requiere ser revisada por un especialista.

- Preparación de los datos

FORMATO

- Los datos obtenidos se organizaron en los siguientes directorios:
 - /resources/Malignos → Clase 0
 - /resources/Benignos → Clase 1
- Una vez organizados los datos, se realizaron pruebas de concepto para revisar la calidad y funcionamiento con un modelo base. Las pruebas consideraban:
 - Imágenes en color o Imágenes en B&W
 - Tamaño de imagen (32x32, 48x48, 96x96, 128x128)
 - Normalización de los datos (/255) o no
- Los mejores resultados se obtuvieron con imágenes **en Blanco y Negro, redimensionadas a un tamaño de 48x48 y normalizadas**, con lo cual este fue el estándar utilizado para las fases posteriores del proyecto.

- AUMENTO DE DATOS

- Para que las clases estuvieran lo suficientemente representadas y obtener un modelo más generalista (disminuyendo así el riesgo de sobre-entrenamiento), se requieren al menos 1000 datos de cada clase.
- Se consideró entonces el aumento de los datos (Data Augmentation) utilizando la clase **ImageDataGenerator** de la librería Keras (de Tensorflow)
- Para el aumento de datos se hizo nuevamente una prueba de concepto para validar la mejor manera de seguir adelante. Las pruebas realizadas fueron:
 - Opción 1: Convirtiendo previamente las imágenes, guardándolas en disco (en la carpeta **/gray/resize**) del directorio de cada clase. De esta manera la operación **Flow** tomaría las imágenes y realizaría el aumento de los datos a partir de ellas.
 - Opción 2: Organizando las imágenes convertidas en subdirectorios de /train, /valid y /test para generar dinámicamente el entrenamiento del modelo, usando la operación de la clase **Flow_from_directory**
 - Opción 3: Organizando las imágenes en subdirectorios de /train, /valid y /test para generar dinámicamente el entrenamiento del modelo, usando la operación **Flow_from_directory** y convirtiendo las imágenes dinámicamente (en memoria).

De las opciones evaluadas, tomando en cuenta que las opciones 2 y 3 necesitan que los conjuntos de Train, Validación y Test estén previamente diferenciados (con lo cual el resultado del modelo ya está condicionado a la selección previa con la que se entrene), seleccioné la **opción 1**. Es decir, **la opción 1 era más flexible al permitir probar el desempeño de la red modificando, entre otras cosas, los tamaños y los conjuntos de entrenamiento, validación y test**. Adicionalmente, es más rápida pues la conversión de las imágenes se hace solo al inicio. Una vez convertidas, se graban en disco y ya son utilizadas para el entrenamiento haciendo que este sea más rápido.

• Entrenamiento y entonación del modelo

Con los datos preparados bajo la alternativa 1, y luego de numerosas pruebas el modelo final seleccionado fue el siguiente:

TRAIN (85%) usando cross validación (20% para validación)

TEST (15%)

EPOCHS = 100 - BATCH SIZE = 48

RED NEURONAL CON LA SIGUIENTE CONFIGURACIÓN (Secuencial)

- Capa Convolutiva con 8 neuronas
- Capa Convolutiva con 16 neuronas
- MaxPool (2X2)
- DROPOUT 25%
- Capa convolutiva con 32 neuronas
- MaxPool (2X2)
- DROPOUT 25%
- FLATTEN
- Capa Dense con 32 neuronas
- Capa Dense con 64 neuronas
- Capa Dense con 2 (Softmax --> salida)

Optimizer='Adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])

Call Backs:

EarlyStopping (monitor='val_loss', patience=10, verbose=1, min_delta=1e-4)

ReduceLROnPlateau (monitor='val_loss', factor=0.1, patience=10, verbose=1, min_delta=1e-4)

Activación = RELU en todas las capas y SOFTMAX en la salida

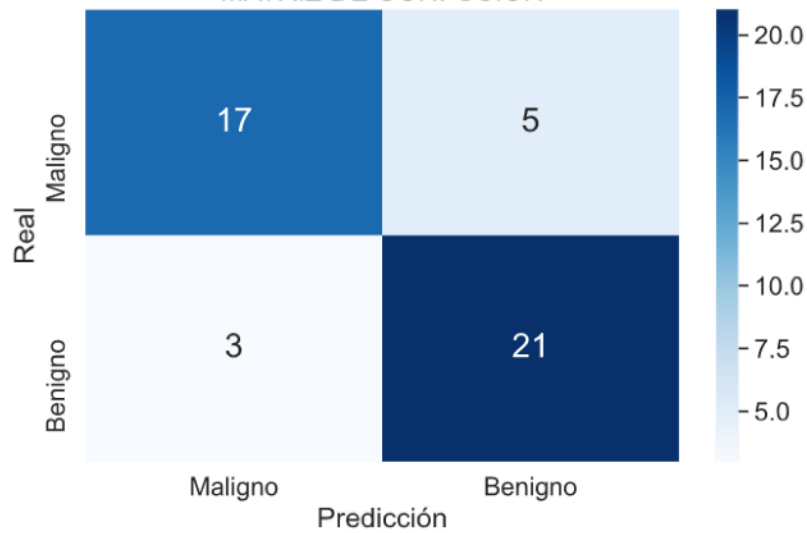
RESULTADOS (luego de 32 EPOCHS): **Accuracy 82,61% - Pérdida 0,4917**

NOTA: El desempeño y resultados obtenidos con las pruebas puede consultarse en el archivo EvaluaciónModelos.xls

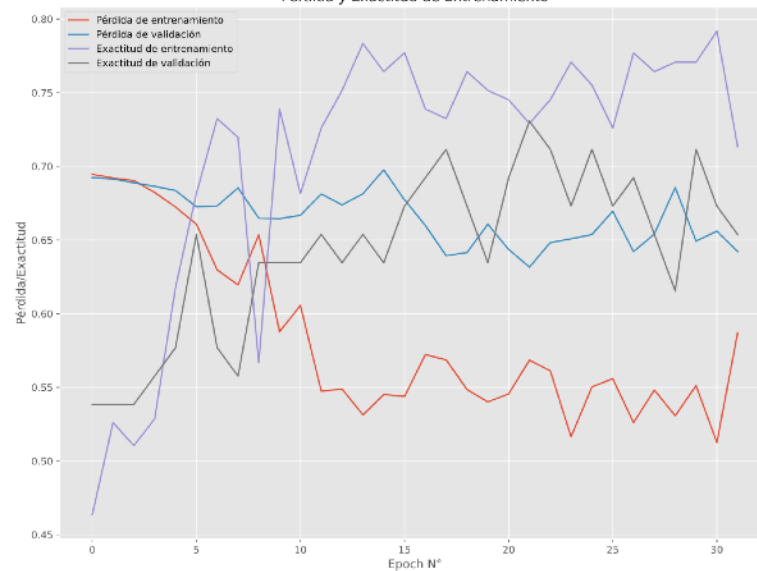
Classification Report

	precision	recall	f1-score	support
Maligno	0.85	0.77	0.81	22
Benigno	0.81	0.88	0.84	24
accuracy			0.83	46
macro avg	0.83	0.82	0.82	46
weighted avg	0.83	0.83	0.83	46

MATRIZ DE CONFUSIÓN



Pérdida y Exactitud de Entrenamiento



Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 46, 46, 8)	80
conv2d_1 (Conv2D)	(None, 44, 44, 16)	1168
max_pooling2d (MaxPooling2D)	(None, 22, 22, 16)	0
dropout (Dropout)	(None, 22, 22, 16)	0
conv2d_2 (Conv2D)	(None, 20, 20, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 10, 10, 32)	0
dropout_1 (Dropout)	(None, 10, 10, 32)	0
flatten (Flatten)	(None, 3200)	0
dense (Dense)	(None, 32)	102432
dense_1 (Dense)	(None, 64)	2112
dense_2 (Dense)	(None, 2)	130
Total params: 110,562		
Trainable params: 110,562		
Non-trainable params: 0		

Una vez seleccionado el mejor modelo, se salvó el mismo (pesos) para poder ser utilizado a través de un API. El API Server provee *endpoints* para dos funciones que permiten, dada la imagen de una lesión:

- 1) Obtener la predicción desde el browser con una imagen seleccionada por el usuario desde un directorio
- 2) Obtener el resultado de la predicción en un archivo en formato json .

• Conclusiones

Tiempo dedicado al proyecto.

El tiempo total dedicado al proyecto fue de 85 horas divididas así

- 19% Factibilidad, planificación y alcance del proyecto
- 30% Obtención y preparación de datos
- 25% Selección y entrenamiento del modelo
- 8% API (Server Flask) para acceder al modelo
- 18% Estandarización y documentación

El tiempo invertido en la calidad de los datos que alimenten el modelo es directamente proporcional a la mejora de los resultados obtenidos en el entrenamiento posteriormente . Es decir, todo el tiempo invertido en jugar con los datos y evaluarlos de distintas maneras fue compensado con los resultados obtenidos posteriormente

Métricas.

En modelos como este, es importante enfocarse en métricas adicionales al accuracy y evaluar los resultados de la matriz de confusión y del reporte de clasificación para decidir cual es el mejor modelo. No necesariamente un modelo con mejor “accuracy” es mejor. En este caso de estudio:

- La métrica recall es más representativa (especialmente el recall en la clase 0 (Maligno). Cuanto mayor sea este valor más casos de lesiones sospechosas esta acertando nuestro modelo por lo tanto es conveniente maximizarlo
- Es sumamente importante tomar en cuenta la matriz de confusión. Los falsos positivos (es decir, predecir que una lesión es Normal -1- cuando en realidad no lo es -0-) son mucho más graves y riesgosos que cuando la predicción falla en el sentido contrario (Falsos Negativos)
- Si tenemos dos modelos con el mismo accuracy, el mejor modelo será, en este caso, es el que obtiene mejor Recall en la clase 0. Es decir, el porcentaje de lesiones Sospechosas que nuestro modelo fue capaz de identificar