Diagnóstico de Cáncer de Piel mediante Red Neuronal Convolucional

Unai López Ansoleaga

29/12/2019

Objetivo

El objetivo del proyecto es crear una red neuronal convolucional para hacer una clasificación binaria (benigno / maligno) de diferentes lesiones en la piel. Esta red debería de ser capaz de detectar los canceres malignos con al menos un 80% de precisión.

Tareas realizadas

- **Instalar dependencias de Python:**
  - Instalar Tensorflow → pip install tensorflow==1.14
  - Instalar OpenCV → pip install opencv-python
  - Instalar Numpy → pip install numpy
  - Instalar tqdm → pip install tqdm
  - Instalar tflearn → pip install tflearn
  - Instalar Pandas → pip install pandas
  - Instalar Matplotlib → pip install matplotlib

- **Descargar datos de Kaggle y adaptar los nombres de las imágenes:**
  - Se han descargado alrededor de 3400 imágenes de canceres tanto benignos como malignos: https://www.kaggle.com/fanconic/skin-cancer-malignant-vs-benign
  - Dado que los nombres de las imágenes no contenían diferenciación entre clases, se han renombrado todos con un script en Python (clase.identificador.png)

- **Equilibrar muestras de imágenes benignas y malignas tanto en la carpeta "train" como en la carpeta "test". En cambio, la carpeta de "validate" no se han equilibrado:**
  - Carpeta "train" → 2272 imágenes (1136 benignos y 1136 malignos)
  - Carpeta "test" → 510 imágenes (255 benignos y 255 malignos)
  - Carpeta "validate" → 367 imágenes (211 benignos y 156 malignos)

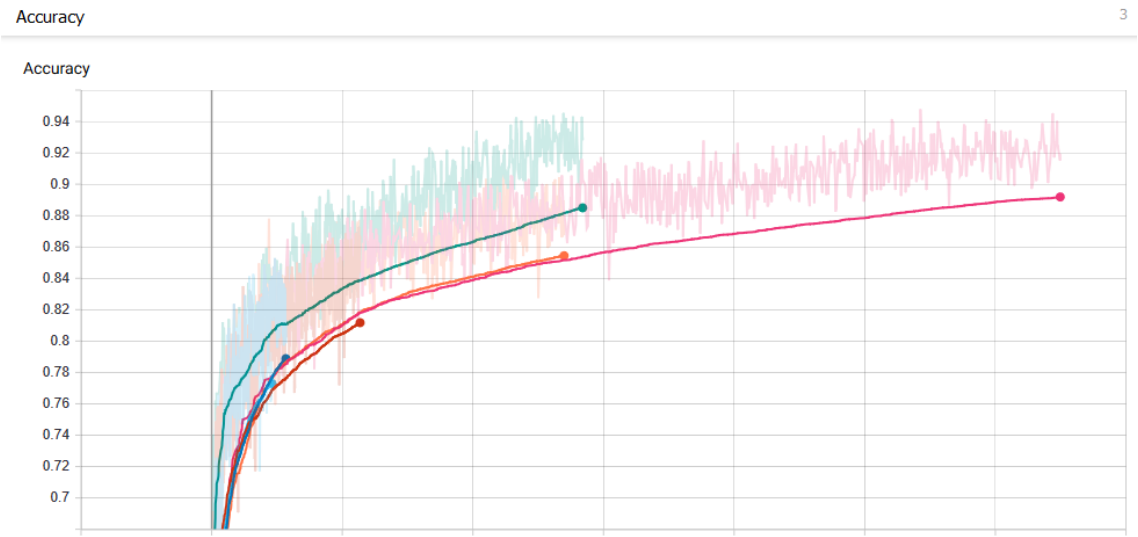
- **Crear red neuronal convolucional**

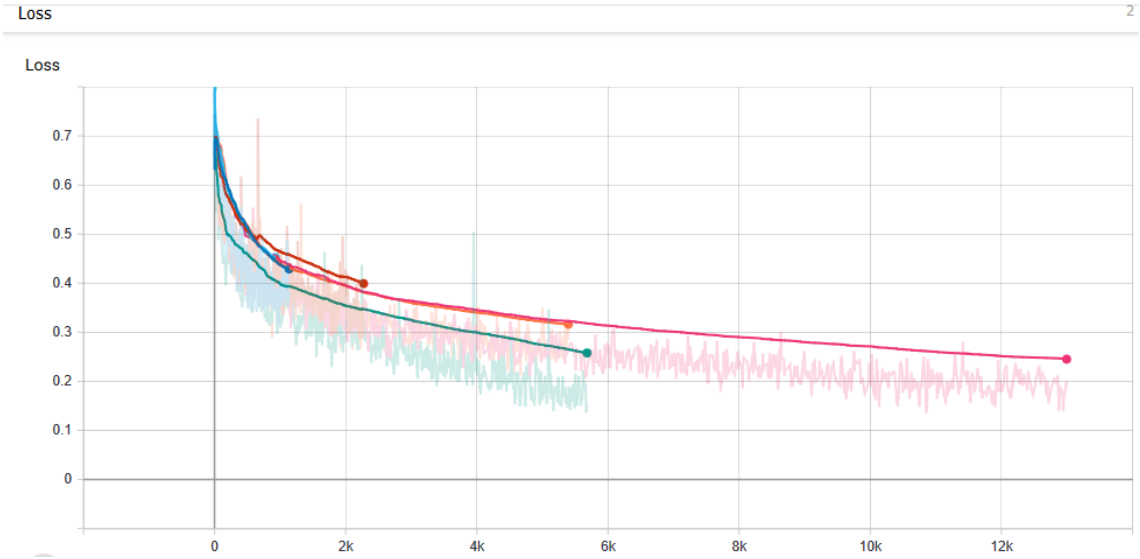
- Mediante un proceso iterativo se han ido probando diferentes parámetros y capas para tratar de optimizar la red. Finalmente, la red ha acabado teniendo 10 capas convolucionales con sus respectivas capas de pooling y 2 fully connected layers. Además de eso, se añadieron algunos dropouts para evitar el overfitting. Todas las capas utilizan la función relu.

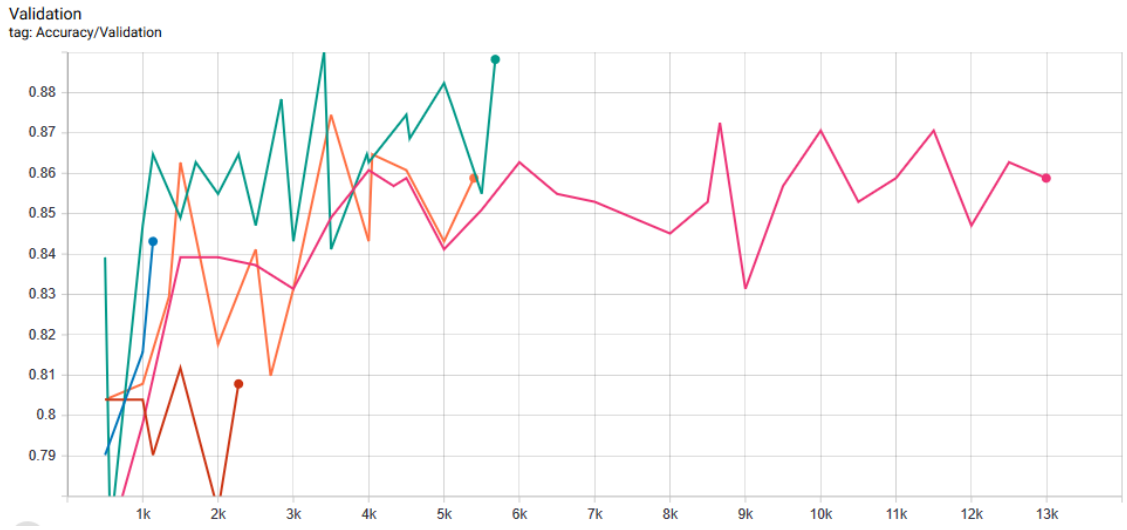
- **Data augmentation**
  - Mediante un proceso iterativo se han probado diferentes tipos de data augmentation. Algunos añadían ruido y no daban buenos resultados. Otros en cambio, mejoraban el rendimiento de la red. Algunos ejemplos:
    - Añadir ruido de fondo (salt pepper noise)
    - Rotar imagen 45, 90, 135, 180, 225, 270 y 315 grados (rotate)
    - Voltear imagen (flip)
    - Mover imagen arriba, abajo, izquierda, derecha, arriba-derecha, arriba-izquierda, abajo-derecha y abajo-izquierda (shift)
    - Aumentar y disminuir brillo de la imagen
  - Se han llegado a multiplicar por 60 la cantidad de imágenes iniciales mediante data augmentation, pero esto no obtenía tan buenos resultados. Finalmente, usando rotaciones de 90, 180 y 270 grados, volteo de imagen, aumento y disminución del brillo y añadir ruido, se ha conseguido multiplicar por 7 la cantidad de imágenes y se han conseguido bastantes buenos resultados.

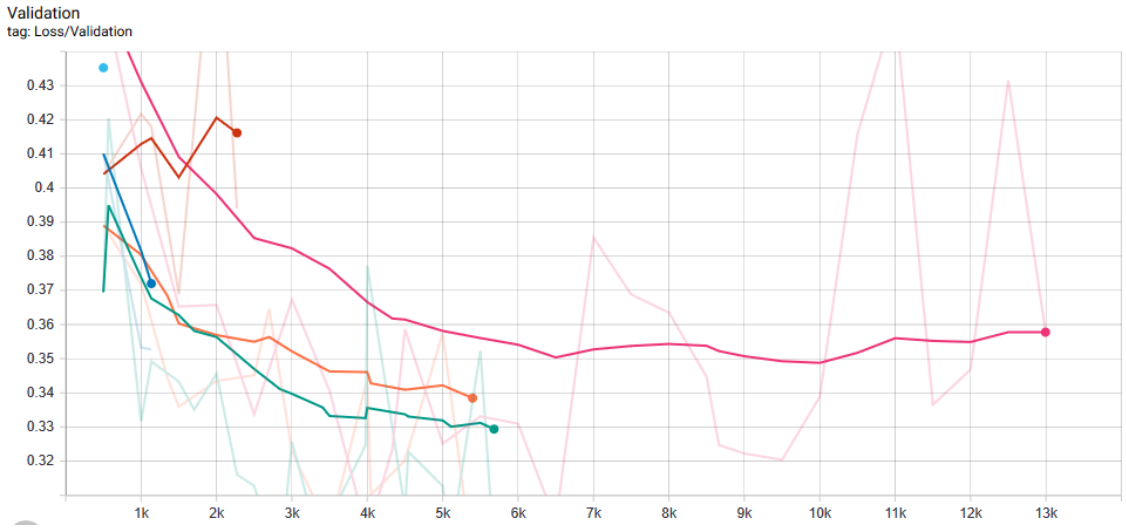
Resultados

Si nos fijamos en la raya verde (el modelo final), es el que mejores resultados tiene tanto en la precisión de los datos de train como en los datos de test. Además, también tiene el mejor resultado en el loss ya que es el que más rápido disminuye.









Sin embargo, la principal razón por la que se ha elegido este modelo a diferencia de los otros, es por los resultados en la matriz de confusión:

REAL / PREDICCIÓN	MALIGN	BENIGN
MALIGN	129	27
BENIGN	17	194

Precisión del modelo según la matriz de confusión:

TOTAL ACCURACY	0.8801089918256131
ACCURACY MALIGN	0.8269230769230769
ACCURACY BENIGN	0.919431279620853

Concluyendo, son unos resultados bastante buenos. Pero dado que es un algoritmo para el diagnóstico médico, la precisión en la detección de los cánceres malignos debería ser mayor. Los falsos negativos en la medicina son mucho más peligrosos que los falsos positivos ya que puede haber vidas en juego.