Inteligencia Artificial
Trabajo Final
Ponderación: 20%
2019-I

Sobre: Deep Learning.
Fecha de entrega: 06/Junio/2019 (jueves)
Grupo: Máximo 3 integrantes
Entrega: Documento + Código + Sustentación en clase.

En el siguiente link pueden encontrar un dataset de clasificación de clasificación binaria mediante imágenes. El problema consiste en clasificar si en imágenes de radiografías de torax se encuentra una neumonía o no. Es necesario crear una cuenta en Kaggle para acceder a la descarga.

Para abordar este problema utilizaremos una aproximación desde el Deep Learning y las redes convolutivas.

A continuación se describen los requerimientos para la solución y su ponderación en la evaluación del trabajo:

1. **Descargar el dataset desde Kaggle (1.2 GB) y explorar (10%)**: Una vez se ha descargado el dataset es importante conocer la estructura del mismo, es decir, la forma en que está organizado. Igualmente se requiere realizar una exploración de las imágenes de cada clase (tamaño, contenido, etc.). Es necesario discutir sobre:
    1. ¿Cuántas imágenes tiene el dataset (entrenamiento y prueba)?
    2. ¿Qué características tienen las imágenes del dataset?
    3. ¿Qué se puede observar en imágenes de cada categoría? ¿Puede usted discernir si una radiografía muestra una neumonía o una condición normal?
    4. ¿Cree que usted que las redes convolutivas podrían ayudar para entrenar un clasificador para este problema? Explique.

2. **Leer el dataset utilizando ImageDataGenerator (10%)**: Debido a que vamos a enfrentarnos a un dataset que no se encuentra en Keras sino en imágenes en carpetas locales, necesitamos una herramienta que nos permita generar el dataset desde disco. En el notebook entregado ya se implementa la lectura del dataset de entrenamiento. Complementar el código para:
    1. Leer el dataset de prueba en la carpeta **test**.

3. **Entrenamiento y Validación (40%)**: En esta etapa usted deberá mejorar el **accuracy** de entrenamiento del clasificador hasta un valor mayor al 97%. Para ello deberá:
    1. Modificar la arquitectura de la red neuronal convolutiva entregada en el notebook en término del **número de capas** y del **número de filtros por capa** ya sea para las capas convolutivas, como para las capas Dense. **En cada caso** documentar el resultado numérico y gráfico del entrenamiento y discutir.
    2. Modificar meta-parámetros de entrenamiento. En particular el learning-rate y el número de épocas.
    3. Evaluar cada caso anterior sobre el dataset de prueba utilizando model.evaluate_generator. Documentar cada caso y discutir.
    4. Guardar en disco cada modelo evaluado: https://machinelearningmastery.com/save-load-keras-deep-learning-models/.

4. **Ajuste técnico (30%)**: En el notebook entregado las imágenes del dataset se redimensionan a un tamaño de 28x28 para poder entrenar el modelo en un equipo de bajas especificaciones. Sin embargo, esta situación tiene un impacto negativo en el accuracy del

modelo entrenado, pues reducir el tamaño tan drásticamente conlleva a una pérdida de información. Para resolver este problema usted deberá:

1. Cambie la re-dimensión de las imágenes a 300x300 en el notebook.
2. Si tiene un equipo con altas especificaciones y que tenga una tarjeta gráfica NVIDIA y una distro de linux, instale y configure Tensorflow + Keras en GPU y ejecute el entrenamiento del mejor modelo de la etapa anterior.
3. En caso contrario, utilizar google colaboratory en entorno de ejecución de GPU. En este caso usted deberá resolver el problema de cargar el dataset de 1.2 GB a google colaboratory. Entrenar nuevamente el mejor modelo de la etapa anterior.

5. **Evaluación final o validación (10%)**: Utilizando el mejor modelo anteriormente entrenado y guardado en disco, evalúe el desempeño del mismo sobre el dataset de validación de la carpeta **val**.

PUNTOS EXTRA (hasta 3/5 puntos): Puntos que se suman a este trabajo o al 20% anterior.

En el mundo del deeplearning existe un concepto conocido como TRANSFER LEARNING. En pocas palabras, este concepto consiste en tomar ciertas capas de una red que ha sido entrenada previamente para una tarea en particular, agregarle nuevas capas propias y entrenar las nuevas capas propias para otra tarea, conservando intactas las capas de la red pre-entrenada. La idea detrás de esto es que uno puede acelerar el proceso de entrenamiento de una solución de deep learning utilizando el entrenamiento que otra persona ya ha realizado para una tarea similar a la nuestra.

Keras cuenta con diversas redes pre-entrenadas que pueden ser utilizadas para realizar transfer learning, por ejemplo, VGG-16/19, Inception y ResNet, entre otras.

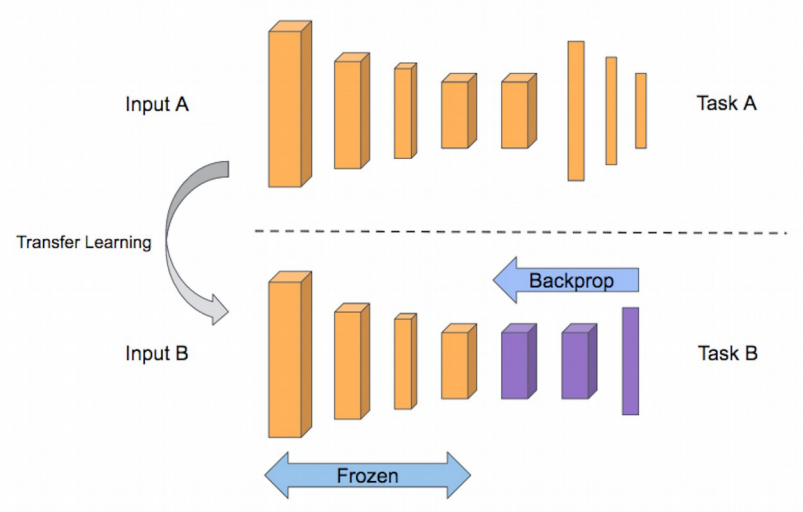

Figura 1: Proceso de Transfer Learning. Tomada de: https://medium.com/@subodh.malgonde/transfer-learning-using-tensorflow-52a4f6bcde3e

Inteligencia Artificial
Trabajo Final
Ponderación: 20%
2019-I

En la Figura 1 se puede observar un ejemplo visual de transfer learning. Suponga que se tiene una red que fue entrenada (red pre-entrenada) para resolver problemas de clasificación de aves (Task A). Dada una nueva tarea (Task B), por ejemplo, clasificar distintos tipos de ranas, el transfer learning de la Figura 1 consistiría en crear un nuevo modelo tomando las primeras 4 capas de la primera red con sus valores de W y b, agregándole 3 nuevas capas, y entrenar este nuevo modelo para el problema de las ranas, manteniendo constante los pesos de las primeras 4 capas y entrenando solo los pesos de las 3 nuevas capas.

En este punto, usted deberá aplicar transfer learning a la clasificación de radiografías antes presentada, entrenar un modelo de este tipo y validar el resultado.