**Ejercicios de redes neuronales densamente conectas y redes neuronales convolucionales**

Para resolver los problemas siguientes, debes acceder a las bases de datos que se encuentran en el siguiente repositorio:

<https://drive.google.com/drive/folders/1q_1GiS_MXf60s9agziMkjamX1MFlNMDj?usp=sharing>

1. Empleando una red neuronal densamente conectada, debes resolver el problema de predecir el precio de vehículos. La base de datos correspondiente se denomina ***“car data.csv”,*** y fue tomada originalmente del sitio Kaggle (<https://www.kaggle.com/nehalbirla/vehicle-dataset-from-cardekho>).

Esta base de datos está constituida por la información de 301 vehículos, donde la información de cada vehículo está representada por 9 atributos, los cuales se enlistan a continuación:

1. Car\_Name
2. Year
3. Selling\_Price
4. Present\_Price
5. Kms\_Driven
6. Fuel\_Type
7. Seller\_Type
8. Transmission
9. Owner

De estos 9 atributos, el número 3, ***Selling Price***, representa la etiqueta de cada una de las instancias.

Tu implementación debe cumplir con los siguientes requisitos:

* Elimina la columna ***Car\_Name,*** ya que sus elementos no son significativos para realizar la predicción del precio del vehículo.
* Sustituye los valores de la columna ***Year,*** por el número de años que han pasado luego de la construcción del vehículo. Por ejemplo, si el valor del atributo ***Year*** de algún vehículo es ***2015***, tienes que reemplazar ese valor por ***6***, ya que ***2021-2015=6.***
* Los valores de los atributos Full\_Type, Seller\_Type y Transmission, están codificados como strings, por lo tanto, es necesario convertir dichos strings a su representación numérica.

Para lograrlo, utiliza la siguiente línea de código de la librería Pandas: df=pd.get\_dummies(data=df,drop\_first=True).

* Tomando en cuenta lo anterior, debes construir dos conjuntos de datos: entrenamiento y validation. El conjunto de entrenamiento debe estar constituido por 210 instancias, mientras que el conjunto de validación debe estar compuesto por 91 instancias.
* Como una guía, considera que las instancias de los conjuntos de entrenamiento y prueba, deben tener los siguientes atributos:

1. Year
2. Present\_Price
3. Kms\_Driven
4. Owner
5. Fuel\_Type\_Diesel
6. Fuel\_Type\_Petrol
7. Seller\_Type\_Individual
8. Transmission\_Manual

* Utiliza la función ***mae*** (Mean Average Error) como métrica de evaluación del desempeño de tu modelo.
* Despliega las gráficas de tu modelo correspondientes a la función de pérdida y a la métrica de evaluación.
* Debes guardar el modelo que tenga el mejor desempeño en el conjunto de validación, para ello, emplea ***callbacks.***

1. Implementa una red neuronal convolucional para resolver el problema de clasificar objetos en 10 categorías distintas. Para abordar este problema, debes utilizar la base de datos CIFAR-10, la cual viene precargada en Keras.

La base de datos CIFAR-10, se compone de 50,000 imágenes de entrenamiento y 10,000 imágenes de prueba. Todas las imágenes de esta base de datos están a color y tienen dimensiones de 32X32. Cada una de las imágenes de los conjuntos de entrenamiento y de prueba, se asocia a una de las 10 clases disponibles. Estas 10 clases se enlistan a continuación:

1. Avión
2. Automóvil
3. Pájaro
4. Gato
5. Ciervo
6. Perro
7. Rana
8. Caballo
9. Barco
10. Camión

El objetivo principal de este problema, consiste en que, al evaluar tu modelo entrenado en las 10,000 imágenes del conjunto de prueba, obtengas una precisión entre 0.70 y 1.0.

1. Implementa una red neural convolucional que resuelva el problema de clasificación del clima. Para resolver este problema, tienes que emplear la base de datos contenida en el archivo ***“Weather\_Dataset.zip”.*** Esta base de datos fue tomada del siguiente sitio web: <https://data.mendeley.com/datasets/4drtyfjtfy/1>

La base de datos en cuestión contiene 4 clases:

1. Cloudy
2. Rain
3. Shine
4. Sunrise

Tu implementación debe cumplir con los siguientes requisitos:

* Construye dos conjuntos de datos: entrenamiento y validación. Al generar estos conjuntos, debes asegurarte de que cada clase del conjunto de entrenamiento tenga como máximo 150 instancias y que el resto de las imágenes de cada clase, se asignen al conjunto de validación, es decir, debes usar todas las imágenes de la base de datos para conformar los conjuntos de entrenamiento y validación. ES OBLIGATORIO QUE LOS CONJUNTOS DE

ENTRENAMIENTO Y VALIDACIÓN LOS GENERES MEDIANTE CÓDIGO, NO PUEDES HACERLO DE FORMA MANUAL.

* Algunas de las imágenes de la base de datos están dañadas, por lo tanto, debes de diseñar una estrategia que te permita solventar este problema mediante código, NO PUEDES ELIMINAR LAS IMÁGENES DAÑADAS DE FORMA MANUAL.
* Debes desplegar las gráficas de precisión y pérdida.
* Debes guardar el modelo que tenga el mejor desempeño en el conjunto de validación, para ello, emplea ***callbacks.***
* Además de utilizar la métrica ***acc*** para evaluar el desempeño de tu modelo en el conjunto de validación, debes desplegar la ***matriz de confusión*** que muestre el rendimiento de tu MEJOR MODELO en el conjunto de validación. También tienes que mostrar el valor de la métrica ***F1*** con respecto al rendimiento de tu MEJOR MODELO en el conjunto de validación.
* Si organizas tus conjuntos de entrenamiento y validación tal como previamente se indicó, y además, empleas las funciones de pérdida y compilación adecuadas, no debes tener mayor inconveniente en que el rendimiento de tu modelo supero una exactitud (***acc***) de 0.80.