

Práctica 3

Aprendizaje Automático

Redes de Neuronas Convolucionales.

Enunciado de la práctica

El objetivo de esta práctica es diseñar, configurar y entrenar un modelo Redes de Neuronas **Convolucionales**.

- La práctica se realizará en los grupos que ya están creados.
- El entregable será el fichero Jupyter Notebook que habréis trabajado en Google Colab) con el que habréis estado trabajando en la práctica. Ahí estará tanto el código como las explicaciones en formato Markdown embebido. Tenéis un útil manual de Markdown en https://colab.research.google.com/notebooks/markdown_guide.ipynb. Os aconsejo que documentéis todo lo que hagáis, aunque sean pruebas que no os hayan salido bien. El objetivo es que contéis una historia. El Jupyter Notebook deberá comenzar con tres líneas:
 - GRUPO <NºGRUPO>
 - Una línea con el Nombre y Apellidos de cada integrante del grupo.
- La práctica deberá subirse a un repositorio de vuestra cuenta personal de GitHub llamada AA_PRACTICA3_GRUPO_NºGRUPO. (Por ejemplo, si sois el grupo 4, vuestra entrega estará en el repositorio de GitHub de alguno de vosotros dos con el nombre AA_PRACTICA3_GRUPO_4). Tenéis que usar el mismo número de grupo que el que usasteis durante la practica 1 y 2. No se admiten entregas fuera de fecha y recordar que GitHub guarda fecha y hora de subida del fichero
- **El enlace a este repositorio deberá estar en el entregable entregado en CANVAS**
- No sólo se evaluará el resultado final sino todo el proceso hasta completarla, la documentación aportada donde se justificarán todas las decisiones de diseño y se explicaciones detalladas de los resultados y su razonamiento.
- **Se evaluará positivamente todo el contenido adicional a la asignatura contenido en la práctica, siempre que guarde relación y aporte valor al objetivo de esta.**
- Es importante demostrar todo el conocimiento adquirido durante las clases teóricas.

Enunciado

Crea un modelo de Red de Neuronas **Convolucionales** que sea capaz de **reconocer y clasificar imágenes en 100 categorías**. Este modelo será **definido, configurado, entrenado, evaluado y mejorado** para posteriormente usarlo para hacer **predicciones**. Para ello tendréis que crear un modelo en **Keras** aplicando de una tirada todos los pasos al conjunto de datos **CIFAR-100 dataset**, que os podréis descargar de <https://www.cs.toronto.edu/~kriz/cifar.html>.

Este conjunto de datos tiene 100 clases que contienen 600 imágenes cada una. Hay 500 imágenes de entrenamiento y 100 imágenes de prueba por clase. Las 100 clases del CIFAR-100 se agrupan en 20 superclases. Cada imagen viene con una etiqueta "fina" (la clase a la que pertenece) y una etiqueta "gruesa" (la superclase a la que pertenece). Aquí está la lista de clases del CIFAR-100.):

Superclass

aquatic mammals
fish
flowers
food containers
fruit and vegetables
household electrical devices
household furniture
insects
large carnivores
large man-made outdoor things
large natural outdoor scenes
large omnivores and herbivores
medium-sized mammals
non-insect invertebrates
people
reptiles
small mammals
trees
vehicles 1
vehicles 2

Classes

beaver, dolphin, otter, seal, whale
aquarium fish, flatfish, ray, shark, trout
orchids, poppies, roses, sunflowers, tulips
bottles, bowls, cans, cups, plates
apples, mushrooms, oranges, pears, sweet peppers
clock, computer keyboard, lamp, telephone, television
bed, chair, couch, table, wardrobe
bee, beetle, butterfly, caterpillar, cockroach
bear, leopard, lion, tiger, wolf
bridge, castle, house, road, skyscraper
cloud, forest, mountain, plain, sea
camel, cattle, chimpanzee, elephant, kangaroo
fox, porcupine, possum, raccoon, skunk
crab, lobster, snail, spider, worm
baby, boy, girl, man, woman
crocodile, dinosaur, lizard, snake, turtle
hamster, mouse, rabbit, shrew, squirrel
maple, oak, palm, pine, willow
bicycle, bus, motorcycle, pickup truck, train
lawn-mower, rocket, streetcar, tank, tractor

Cuestiones para implementar y responder

- Debéis decidir la estructura de red neuronal convolucional (número de parámetros) y los valores de todos los hiperparámetros que permitan tener la mayor precisión posible en los datos de test.
- Explicar la salida de la llamada `model.summary()` de vuestra red definitiva.
- Seguir los modelos metodológicos vistos en clase (sobre un modelo base, ir incrementando la complejidad para mejorar el modelo hasta donde se pueda)
- Ver los conceptos de `batch_normalization` y `dropout` y ver si se podría mejorar el modelo con ello.

https://keras.io/api/layers/normalization_layers/batch_normalization/
https://keras.io/api/layers/regularization_layers/dropout/

- Ver los conceptos de callbacks (Decaimiento del ratio de aprendizaje) y ver si se podría mejorar el modelo con ello.
https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/LearningRateScheduler y <https://keras.io/api/callbacks/>

Memoria

- Es necesario generar una memoria en la que recogeréis los distintos pasos que habéis realizado para resolver la práctica.
- La memoria debe estar correctamente redactada, sin faltas de ortografía y contener puntos como la introducción y las conclusiones, entre otros.
- Debéis explicar cómo habéis afrontado la solución de la práctica, cómo se ha dividido el trabajo y cómo se ha gestionado el equipo para trabajar en ella.
- Podéis adjuntar imágenes en la que deseáis mostrar algún comportamiento particular o relevante que os haya surgido.

IMPORTANTE: Todo esto, como hemos comentado anteriormente, estará integrado en el Jupyter Notebook dentro de Google Colab

Entregables

El único entregable subir a la tarea de Canvas es el siguientes:

- El enlace de GitHub (el de la persona del grupo a través del cual queráis tener el código de la práctica) diciendo a que grupo pertenecéis

La fecha tope de entrega de esta práctica es el viernes día 1 de diciembre a las 23:59.