

Ejercicios clase CNN

Redes Neuronales Convolucionales

1. Responde a las siguientes preguntas.
 - a. Si a una imagen 40x40 le aplicamos un maxpooling con filter size = 2 y stride =2 que dimensiones quedan?
 - b. Si a una imagen 40x40 le aplicamos un max pooling 5x5 y stride 1, que dimensiones quedan?
 - c. Si a una imagen 7x7 se le aplica una convolución 3x3 con stride = 1, que dimensiones quedan?
 - d. Si a una imagen 7x7 se le aplica una convolución 3x3 con stride = 2, que dimensiones quedan?
 - e. Si a una imagen 6x6x3 le aplicamos una convolución 3x3x3 qué dimensiones quedan? Y si le aplicamos dos convoluciones 3x3x3?
 - f. Si tienes 10 filtros convolucionales 3x3x3 en una red neuronal con una activación ReLU, cuantos parametros tienes?
2. Carga el data set de Aerial Cactus Identification. Es un dataset de imágenes para identificar si en la imagen hay un cactus o no. Tendremos una carpeta con imágenes y un csv, vamos a transformar esas imágenes a un dataset. **Posibles ideas:**
 - a. Lee con pandas el archivo train_cactus.csv. Que tiene este dataset? Para leer las imagenes hay que recorrer una columna del dataframe.
 - b. Estudia el módulo image del módulo tensorflow.keras.utils. Utiliza la función load_img para cargar una imagen del dataset cualquiera.
 - i. Del mismo módulo anterior estudia la función img_to_array. Transforma la imagen cargada en un array.
 - c. Escala el array entre 0 y 1 dividiendo por 255.
 - d. Creando diferentes listas, carga todas las imágenes en una lista de arrays.
3. Crea una red neuronal con:
 - a. 1 hidden layer convolucional y 32 neuronas y de tamaño 5x5.
 - b. Aplica un Max Pooling 2x2, con padding = 'same'.
 - c. 1 hidden layer convolucional y 64 neuronas y de tamaño 3x3.
 - d. Aplica un Max Pooling 2x2, con padding = 'same'.
 - e. 1 hidden layer convolucional y 64 neuronas y de tamaño 3x3.
 - f. Aplica un Max Pooling 2x2, con strides = 2.
 - g. Transforma los datos a un vector *flattened*.
 - h. 1 hidden layer con 100 neuronas.

- i. 1 último layer que devuelva la probabilidad de que haya un cactus en la imagen.
- j. Que dimensiones tiene los inputs y outputs de cada layer?
- k. Entrena esta red neuronal.
- l. Qué métricas obtienes en el test_set?
- m. Haz un plot con la evolución por epoch de la performance del modelo.
- n. Repite el modelo añadiendo dropout y regularización, donde deberías añadir el dropout?
- o. Añade el batch normalization después de cada layer convolucional, utilizando `model.add(BatchNormalization())`