EXERCISES WEEK 8: Convolutional Neural Networks with Tensorflow&Keras

Use these guidelines in combination with the notebook “CIFAR10_CNN.ipynb”:

Warm-up:

Get familiar with the idea of convolutions on images by experimenting with the following interactive links:

- https://poloclub.github.io/cnn-explainer/
- https://setosa.io/ev/image-kernels/

Exercise 1- Code Convolutions:

Choose a picture and compute a convolution on it, by implementing a e.g. vertical filter.

Exercise 2: CNN for image classification with CIFAR10

In this exercise we will apply the same code to the CIFAR10 Dataset. (https://www.cs.toronto.edu/~kriz/cifar.html). This also contains 10 categories of pictures, but in this case we deal with more than numbers and RGB-images (Format, 32x32px, 3 color channels)

This dataset is available to be loaded directly from Tensorflow:

- https://www.tensorflow.org/api_docs/python/tf/keras/datasets/cifar10

a) **Implement with the Cifar Dataset a Convolutional Neural Network,** by adding on top of your previous architecture as many Conv2D and MaxPool Layers (see lessons slides) as you think it is best. Try to optimize the performance, by checking with TensorBoard.
Better?
How are the training vs validation performance?
How is the performance wr.t. the ANN Architecture used for MNIST?

To improve performance consider including: Dropout, EarlyStopping, Regularization, Adam-Optimizer…

b) _Once you are happy with your training and validation, **select from internet real image belonging to the same categories which appear in the CIFAR10 Dataset and try to test if your model can recognize them correctly.**

To correctly load the image, the following functions will be useful

- tf.keras.utils.load_img -> to load your image. Be aware that you need to set the target_size to rescale the images to the necessary resolution (32x32)
- tf.keras.utils.img_to_array(img)
- tf.expand_dims(img, 0) -> to add the image to a batch where it's the only member.

c) Try now to **select an image from a category NOT part of the CIFAR10 Dataset** (e.g. house). What will happen? In which category will the picture be classified? What could be a way to have a more elegant solution in this case? Discuss with your colleagues!