Practice 4, 2022

Convolutional Networks

Problem

Malaria is a deadly, infectious mosquito-borne disease caused by Plasmodium parasites. These parasites are transmitted by the bites of infected female Anopheles mosquitoes. With regular manual diagnosis of blood smears, it is an intensive manual process requiring proper expertise in classifying and counting the parasitized and uninfected cells. Typically this may not scale well and might cause problems if we do not have the right expertise in specific regions around the world. We are lucky to have researchers at the Lister Hill National Center for Biomedical Communications (LHNCBC), part of National Library of Medicine (NLM) who have carefully collected and annotated this dataset of healthy and infected blood smear images. There is a balanced dataset of 13779 malaria and non-malaria (uninfected) cell images.

In the malaria.zip file in Atenea campus you will find near thousand images downloaded from the official website that have been resized (64x64x3). The images are organized in three folders: train, validation and test. Deep Learning models, or to be more specific, Convolutional Neural Networks (CNNs) have proven to be really effective in a wide variety of computer vision tasks. Here we will apply CNN to build a classifier that allows us to discriminate images from uninfected and infected patients. We ask:

1. Implement a CNN according with the summary in Figure 1.
2. Define conveniently the model (optimization, loss, metric, ...).
3. Using `tfruns` package tune the hyperparameter `batch_size` exploring the grid $c(16, 32, 64)$.

4. Implement an early-stopping `callbacks()` to interrupt training when validation accuracy stops improving for more than two epochs.

5. Assess the performance of the CNN predicting the categories of test images and obtain the confusion matrix.

6. Implement a Convolutional Autoencoder (CAE), with 10 nodes in z layer (or bottleneck). Feel free to choose the number of convolutional layers, filter sizes, number of filters, ...

7. Represent graphically the results from images test to show the association between z layer activations and the class images.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 32)	896
max_pooling2d (MaxPooling2D)	(None, 31, 31, 32)	0
conv2d_1 (Conv2D)	(None, 29, 29, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_2 (Conv2D)	(None, 12, 12, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 128)	0
conv2d_3 (Conv2D)	(None, 4, 4, 128)	147584
max_pooling2d_3 (MaxPooling2D)	(None, 2, 2, 128)	0
flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 512)	262656
dense_1 (Dense)	(None, 1)	513

Total params: 504,001
Trainable params: 504,001
Non-trainable params: 0

Fig. 1: cnn summary