Automatic classification of galaxies using the Galaxy Zoo data and supervised learning

Supervision: Pedro Cunha and Ana Paulino-Afonso – IA and UPorto

Description: Galaxies can be classified by their morphology, as proposed by Hubble in 1936. The galaxy's shape is correlated with its physical properties, such as star formation rate and gas fraction, and also with its merger history (for example, the existence of mergers with other galaxies), and gas fraction. These correlations are a vital step towards a better understanding of the formation and evolution of galaxies. The scientific and technological advancement of observational instruments increased substantially the number of detected sources, making it impossible for astronomers to visually classify galaxies one by one. The Galaxy Zoo project is a community-based program for the visual classification of galaxy images obtained with the Sloan Digital Sky Survey (SDSS), where people were asked to participate and help to classify 900,000 galaxies into elliptical, spiral and merger products. This provides the perfect environment for the application of automatic classification algorithms using machine learning to further classify sources to be detected with new instruments, such as ESA's Euclid space telescope. In this project, you will build a machine learning pipeline capable of identifying galaxy morphology with image data using the TensorFlow platform. No past experience with machine learning frameworks is necessary, but practical knowledge of Python is required.

Expected Output: The main objective of this project is to showcase the usefulness of Machine Learning techniques for scientific research, in particular in astronomy. The student is expected to build a Python notebook (Jupyter Notebook, JetBrains DataSpell, etc) capable of processing galaxy images and providing morphological classifications.

Detailed Plan:

Task 1: Exploring Galaxy Zoo data

The Galaxy Zoo project provided the classification results in the following website: https://data.galaxyzoo.org/

The first thing to do is download the data. You will get the dataset from Galaxy Zoo 2. Read this page carefully alongside this one. It is important to cross-reference the

images with the classification from Galaxy Zoo. You can do it by using the ObjID. The class you will consider for the classification is the “gz2_class”.
My recommendation is for you to identify the classes in the dataset and select a random sample of sources with that label (e.g. 2,000 galaxies classified as Er, etc). You are free to choose the number of classes you want to use (e.g, 2 for a binary classification, or all of them). Remember that the number of chosen classes will increase the size of the data set and the computation processing time.
At the end, you should have a main folder with subfolders that corresponds to the classes of the galaxies. This will be helpful for later!

Task 2: Preparing the pipeline

After the data processing task, you need to start building the pipeline.
Here I propose you check the following examples:

- https://www.tensorflow.org/tutorials/keras/classification
- https://towardsdatascience.com/create-image-classification-models-with-tensorflow-in-10-minutes-d0caef7ca011
- https://medium.com/edureka/tensorflow-image-classification-19b63b7bfd95

This is an experimental task, which means you will do a lot of things by trial and error. It is important you understand the different steps and how they are relevant for your data set.

Task 3: Testing your model

After you have your model ready, it is time for evaluation, since we are doing a supervised task. This task is actually pretty linked with the previous one. You should build at least 2 models: (1) Baseline: this should be a simple one for comparison and to understand how complexity help the problem in hand; (2) CNN: Taking into consideration the model (1), you can try to add more layers to the deep learning model, in particular 2D convolutional layers. You are encouraged to test multiple models and achieve the best result possible.

Have fun!

	Day	Plan	Expected tasks
Monday	11	Introduction	Introduction
Tuesday	12	Work/Follow-up meeting	Task 1
Wednesday	13	Work	Task 1
Thursday	14	Work/Follow-up meeting	Task 2
Weekend			
Monday	18	Work	Task 2
Tuesday	19	Work/Follow-up meeting	Task 2
Wednesday	20	Work	Task 2
Thursday	21	Work/Follow-up meeting	Task 3
Weekend			
Monday	25	Work	Task 3
Tuesday	26	Work/Follow-up meeting	Task 3
Wednesday	27	Work/Follow-up meeting (if needed)	Task 3