The Galaxy Zoo Challenge

Machine Learning project-report by Mattia Sirressi

Stockholm University

January 6, 2020

1 Introduction

In this protocol I report on my first application of machine learning techniques to real data, as part of the course assignment. Being an Astronomy PhD student in the Galaxies group, I was mostly interested in the available projects related to extragalactic astronomy. Ultimately I opted for the project about galaxies morphology.

One of the fundamental scientific questions of all times is: why and how are we here? Or more precisely what is the structure of the Universe and how does it evolve? The galaxies are the large structures of the Universe: we count 100 billions of them[1] and they appear in many shapes, colors, content: dark matter, stars, gas, dust. However, what is not yet completely understood is how they form and evolve. To answer this questions we can start by classifying their morphology simply by looking at their optical images.

1.1 The Galaxy Zoo Challenge

The Galaxy Zoo is a "citizen science" crowdsourcing project where users were asked a sequence of questions aimed at classifying tens of thousands of galaxies in classes and sub-classes[2]. The project is described in details in the paper Willet et al. 2013. Figure 1 shows the structure of the decision tree used to classify the galaxies with multiple-choice questions. The result of this project is a set of probability distributions for the classes and sub-classes of each galaxy.

The Galaxy Zoo Challenge is an online competition on the Kaggle website[3] held in 2013, the same year the article was published. The challenge consists in analyzing the same galaxies images of the citizen-scientists project in order to build an algorithm that is able to reproduce the closest probability distributions to the ones determined by the human classification.

My project consisted in applying some of the machine learning techniques learned during the course on the data available for the Galaxy Zoo Challange.

2 Preparation of the data

2.1 Description of the data

Below a schematic description of the data available and relevant for my project:

- images_training: a folder with jpg images of 61578 galaxies, files are named according to their GalaxyId.
- solutions_training: a csv file with the probability distributions (over a total of 37 answers) representing the human classification of all the 61578 galaxies

[1] the same number of stars in one galaxy

[2] For example one class is constituted by the smooth galaxies, one sub-class is constituted by the smooth cigar-shaped galaxies

[3] https://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge/overview

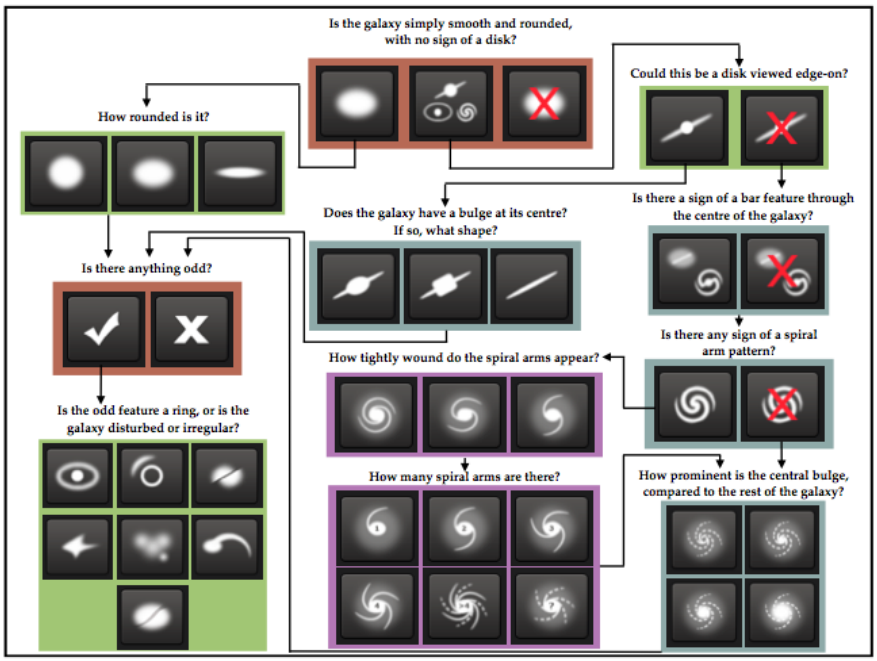

Figure 1. Flowchart of the classification tasks for GZ2, beginning at the top centre. Tasks are colour-coded by their relative depths in the decision tree. Tasks outlined in brown are asked of every galaxy. Tasks outlined in green, blue, and purple are (respectively) one, two or three steps below branching points in the decision tree. Table 2 describes the responses that correspond to the icons in this diagram.

Figure 1: Decision tree of the Galaxy Zoo project, taken from Willet et al. 2013.

In the picture of a machine learning task, the data points x are the images of the galaxies whereas the labels y to be predicted are the probability distributions. The images are 424x424 pixels and each pixel has 3 values (RGB colors), therefore the data points consists in arrays of dimension 424x424x3 = 539328; while the total number of classes and sub-classes is 37, thus the labels consist in arrays of dimension 37.

2.2 Plotting the data

The first step was to import the jpg images in python and convert them into *numpy* arrays. This required a brief research both on google and on the code of the best ranked participants of the challenge, which is available on the Kaggle website. I chose to use the *skimage* library and its package *io* to import the jpg images into arrays that can be plotted using *matplotlib*. Figure 2 shows one of the galaxy plotted with this procedure.

The second step is to visualize the human classification of the galaxies, which is stored in the csv file mentioned above. The first column of this file gives the GalaxyId and the following 37 columns determine the probability of each class or subclass. It is extremely important to notice that these numbers cannot be treated as probabilities of the same distribution because there is a complex decision tree behind the classification: for example the first three classes (smooth, disk, artifacts) already constitute a complete classification meaning that their numbers sum up to 1. Therefore the total sum of the probabilities of each galaxy will always exceed 1.

In order to read the human labels I opened the csv file and read it line by line, saving in a list all the galaxies Ids, which is important to later import all the images of the galaxies and match them with their corresponding probabilities according to the GalaxyId written in their name. Figure 3 shows the human classification of one galaxy, namely the probabilities across the 37 classes and sub-classes.

3 Choice of the task and method

It is worth stressing that the Galazy Zoo competition is about analyzing the jpg images of galaxies to find automated metrics that reproduce the probability distributions derived from human classifications. It is not about finding a classification algorithm that can determine the true morphology of a galaxy

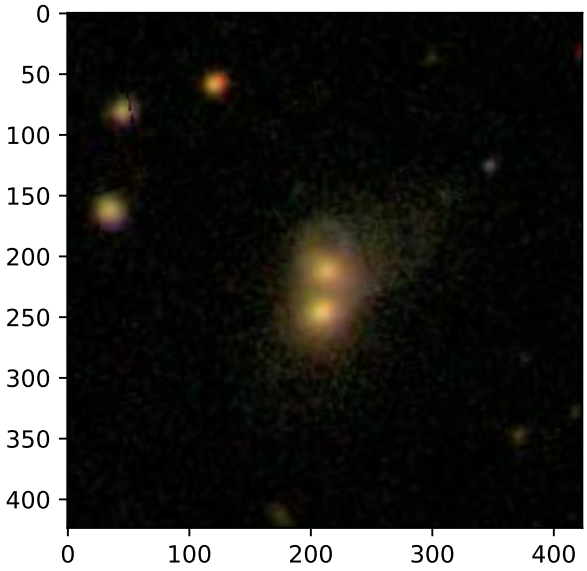

Figure 2: First plot-test of the galaxy with GalaxyId = 100561

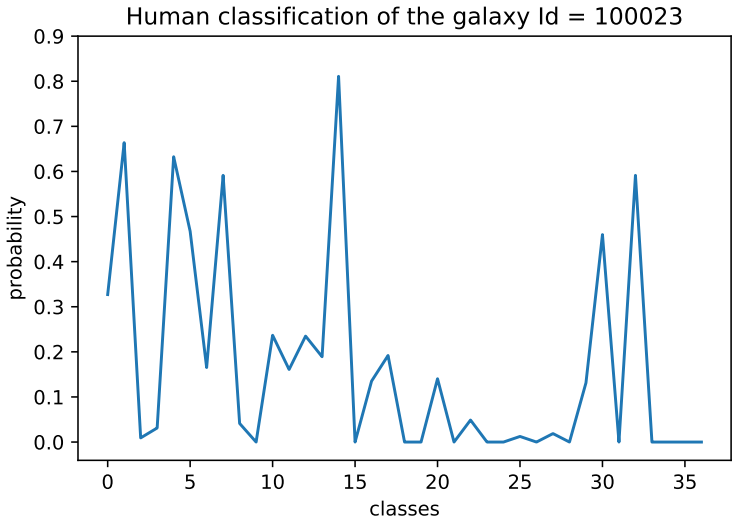

Figure 3: Human labels of one galaxy determined during the crowdsourcing project

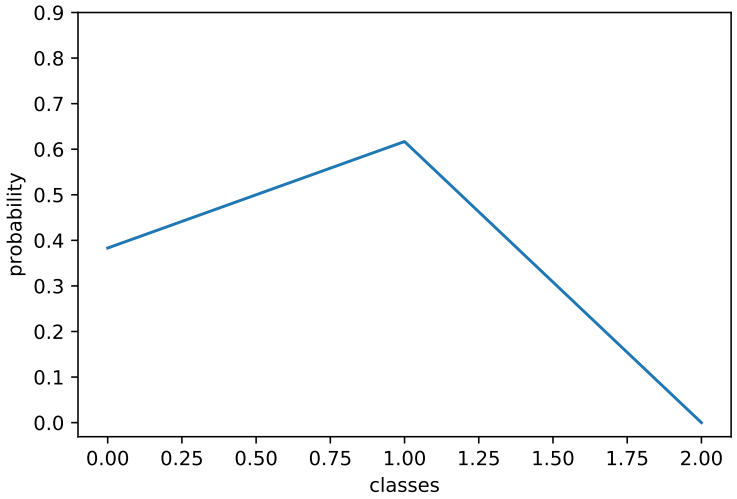

Figure 4: Probability distribution over the three classes: 0 = 'smooth', 1 = 'disk', 2 = 'artifacts'

better than human can do. It is about finding an automated algorithm that reproduces the same (or closest) labels as determined by humans. I make this point here at the beginning because it is extremely important, before proceeding with any machine learning task, to understand what kind of problem I want to solve. The complex structure of the decision tree prevents you to treat this problem as a simple classification task, it is rather a multiple-classification task meaning that for each question of the decision tree the answers give a different classification of the galaxies. By doing some research on the discussion section of this competition on the Kaggle website, I concluded that the competition is a regression task: what we want to fit is a multi-variable function whose output is a vector of 37 numbers predicting the probability distributions.

Before attempting this with a Neural Network (NN) I decided to solve a simplified problem in order to deal with different types of tasks during this educational project. I transformed the Galaxy Zoo challenge in a simple classification problem restricting to the first three classes[4]:

- smooth galaxies (without features)
- disk galaxies (or with features)
- artifacts (or stars)

Figure 4 shows the probability distribution of the three classes in one of the galaxies.

This specific task is a multi-class classification with soft labels (not binary vectors, i.e. it is not certain that a galaxy belongs to one of the three classes). For simplicity I changed the soft labels into hard labels, which is I treated the class with maximum probability as the true one to be predicted by my classification algorithm. This simplification would be legit if I only chose those galaxies that most certainly belong to one class, say one probability is larger than 80%. However, I treated this simplified problem only briefly because I focused more on the regression problem later, which is actually what the Galazy Zoo competition is about.

4 A simplified classification problem

I decided to try two different methods for the simplified classification task: logistic regression and classification with CNN. I expected the logistic regression to be less accurate because this morphology classification is a highly non linear problem. Moreover, since the images are quite large (424x424 pixels, 3 pixel values) I decided to use only a fraction of the labelled galaxies: 800 images as the training data set, 100 as validation data set and 100 as test data set.

[4]Remember that the probabilities of these three classes sum up to 1

4.1 Multi-class logistic regression

First of all I scaled the data to have zero mean and unit variance, which is required by the regressor. I applied the logistic regressor with 'sag' solver and 'l2' penalty. Then I fit the data using the hard lablels, binary vectors indicating which of the three classes is most likely the true one.

Logistic Regression took 30 seconds for 800 training points and gave an accuracy of 72%. The validation data set has been used for estimating the accuracy. Also, I noticed that the labels are quite unbalanced: most of the training points belong to class 0 (smooth) or 1 (disk), i.e. most of the images represent galaxies and not artifacts. I did not understand why the sparsity, that should be the percentage of non-zero weights in the model, has a zero value.

4.2 Classification with a convolutional neural network

I wanted to construct a NN because it is typically the best solution for problems involving images (highly non-linear). Furthermore, it is smart to use filters (convolutional layers) to detect patterns in the images, e.g. spiral arms, bars, round shapes, edge-on disks, bulges, rings, excetera... and therefore I chose to build a Convolutional Neural Network (CNN).

First of all I applied normalization to the data, which generally leads to a faster learning/convergence. I designed one of the simplest model using *tensorflow.keras*: a linear style neural network with the structure described below.

- 1st convolutional layer with 10 filters 7x7
- 1st pooling layer to coarse-grain the images by a factor of 4 both sides
- 2nd convolution layer with smaller filters 5x5
- 2nd pooling layer to coarse-gran the images by a factor of 2 both sides
- dense layer with 128 neurons
- last layer with 3 possible outputs

Before adding the dense layer I needed to flatten the input data in 1-D arrays. For the last layer I chose 'softmax' as activation function. After designing such NN I compiled it using the 'sparse-categorical-crossentropy' as loss function, and 'adam' as optimizer, which is a variant of Stochastic Gradient Descent. I ultimately fit the model running 4 epochs.

The CNN took 20 seconds per epoch with 800 training points. I evaluated the accuracy on the validation data set obtaining 75%, which is a few percent better than the accuracy of the logistic regressor. The performance of the CNN can be in principle be improved adding more convolutional layers and using more epochs. Both the logistic regressor and the CNN can be improved with the following modifications:

- use more training images (but this will imply higher computational cost)
- crop the images (because the galaxy appears always in the central part of the image), which can reduce the computational cost and in principle improve the fit since meaningless parts of the images are discarded
- use weights for imbalanced classes

5 The Challenge: a regression problem

I used a similar CNN to the one described above for reproducing the probability distributions of all classes and sub-classes, which is what the challenge asks for. Because the galaxies take only a small area in the centre, as already mentioned above, what is outside the central square 212x212 is not discriminative. For this reason I cropped the images with a central square of size 212x212. Figure 5 shows the cropped image of one galaxy.

A previous test of regression with CNN shows that cropping does not significantly improve the performance of the fit, nonetheless it is crucial for reducing the number of parameters of the model and cut

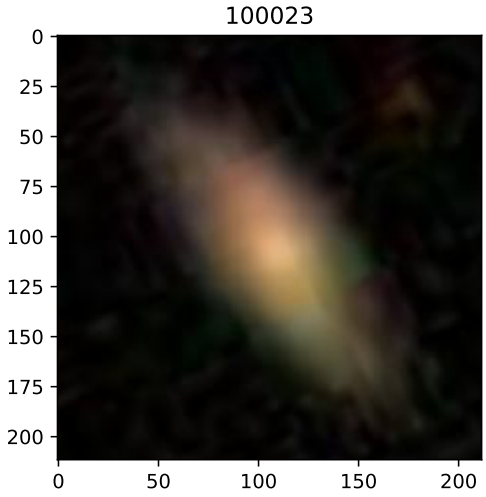

Figure 5: Cropped image of one galaxy, the smaller size contains all the meaningful features

down the computational cost. This allows to use larger data set, increasing it by a factor of 10: 8000 images as the training data set, 1000 as validation data set and 1000 as test data set[5].

I designed a CNN which has basically the same structure as the one used for classification except I accounted for the different size of the images, the different number of neurons in the last layer (37, the number of probabilities to predict). For compiling the model I used the Huber loss function, which is a good choice for multi-variable regression because is not sensitive to outliers and has no problem with the non-smooth nature of the origin. Another important difference is the metrics by which I measured the performance of the model during its training. We cannot talk anymore of accuracy because we are not solving a classification task, instead we need to measure the distance between the probabilities given by the human and those predicted by the model. I chose the metrics used in the Galaxy Zoo Challenge to determine the ranking of the participants, i.e. the Root Mean Squared Error (rmse).

CNN took 80 seconds per epoch with 8000 training points. I evaluated the rmse on the validation data set obtaining 0.22, which is significantly higher than the scores obtained by the top 50 ranked participants, ranging from 0.07 to 0.1. Figure 6 shows how the predicted 'distribution' of probabilities differs from the one determined by humans. The maximum value among the 37 probabilities is well predicted but all the rest of numbers are scarcely predicted.

Again, the performance of the CNN can be in principle improved using more data, adding more convolutional layers and using more epochs, which unfortunately requires longer training time. One easy solution to this would be to use COLAB, a jupyter notebook environement offered by Google, that runs on a freely rentable GPU.

6 Difficulties encountered

- Familiarity with python, lists, numpy arrays, float-type. In the first part of the project I made slow progress because of my little knowledge of python. In particular I needed some time to understand the difference and its importance between lists and numpy arrays.
- Classification or regression? After I could load and plot the data I was deeply thinking of what kind of task I was asked (or I wanted) to solve. The Galaxy morphology project sounds naturally as a classification task but a better look into the challenge made clear it is a regression problem.
- Solo project. The possibility to discuss about a project with a team mate is fundamental because it can help choosing a good approach, recognizing and avoiding errors, solving little implementation problems. Doing the project on my own, I missed all of the above.

[5]Strangely this did not reflect a significant improvement of the score

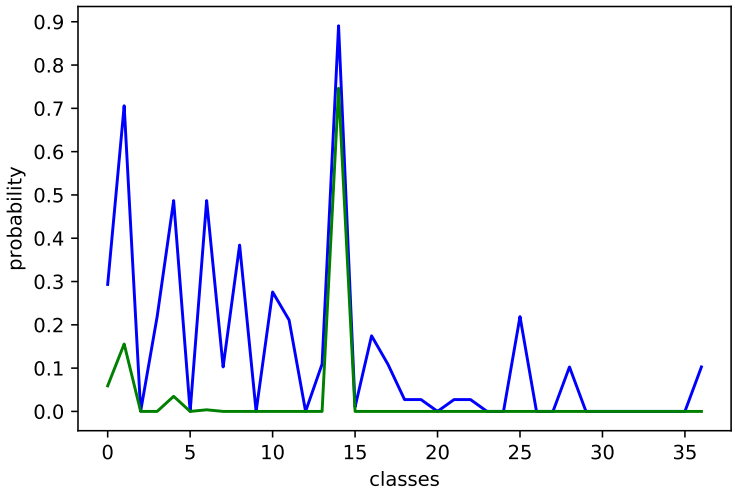

Figure 6: In green the distribution predicted by the CNN model, in blue the distribution produced by the humans

- Time. The limited amount of time I could spend studying the course material and applying ML techniques certainly prevented this project to be thorough.