Read the Docs Template Documentation

Release 1.0

Read the Docs

Aug 29, 2017

Contents

This code uses convolutional neural networks (with tensorflow) to estimate the parameters of strong gravitational lenses. Unfortunately we're not very good at coding, so you'll find that the code is messy, not well documented, and crazily written. However, that shouldn't discourage you from trying it out. Because it's a pretty pretty cool thing: The code can recover the parameters of gravitational lenses in a fraction of a second. Something that used to take hundreds of hours!

The results of this work have been published in a Nature letter "Fast Automated Analysis of Strong Gravitational Lenses with Convolutional Neural Networks" (Hezaveh, Perreault Levasseur, Marshall, 2017) and another paper, "Uncertainties in Parameters Estimated with Neural Networks: Application to Strong Gravitational Lensing", submitted to the Astrophysical Journal Letters (Perreault Levasseur, Hezaveh, and Wechsler, 2017). In the next few months we'll be slowly making this code more user-friendly and extend it to more interesting and complex lensing configurations.

CHAPTER 1

Installation

You will need to have a working version of python. You'll also need the following libraries/packages:

1. tensorflow (instruction here: https://www.tensorflow.org/install/)
2. PIL
3. scipy/numpy
4. matplotlib (if you'd like to generate plots in the ipython notebook)

Once you have these, you can simply clone the repository with git:

```
$ git clone git@github.com:yasharhezaveh/Ensai.git
```

Then, you'll need to download these data files:

1. The trained network weights
2. a sample of lensing images to demonstrate the tool
3. a few cosmic ray and artifact maps

After downloading these either double click on them in Mac, or untar them from the commandline with:

```
$ tar xvfz CosmicRays.tar
$ tar xvfz SAURON_TEST.tar
$ tar xvfz trained_weights.tar
```

Make sure that the unpacked folders are inside "data/" folder in Ensai. Now you should be able to run the ipyn notebook or play with the python scripts!

CHAPTER 2

Overview of the code

Here you will find a quick summary of the codes provided.

You can start your python environment and execute "init.py". This file sets up the necessary variables to perform various neural network tasks (e.g., training or testing). Once this code is executed, nothing happens. You can now run "train.py" to train a model, or run "single_model_predictions.py" to get the predictions of a particular neural net for the test/validation samples.

If instead of "single_model_predictions.py", you run "combo_prediction.py", you will get the predictions of 4 different networks, combined together. This will require that you have the weight files properly stored in your data/trained_weights folder (see installation).

The models (with the exception of Inception.v4) are defined in "ensai_model.py". It is quite easy to modify these or to add new models.

We use "get_data.py" to produce simulated images. This file contains a number of functions (e.g., apply_psf, add_poisson_noise) which are applied to previously saved simulated images of gravitational lenses. the function "read_data_batch" reads the noise-free images from the disk, and applies these effects to them.

CHAPTER 3

Citation:

If you use this code for your research please cite these two papers:

1. "Fast Automated Analysis of Strong Gravitational Lenses with Convolutional Neural Networks" (Hezaveh, Perreault Levasseur, Marshall, Nature, 2017)
2. "Uncertainties in Parameters Estimated with Neural Networks: Application to Strong Gravitational Lensing" (Perreault Levasseur, Hezaveh, and Wechsler, ApJL submitted, 2017)