A Short Instruction for nnbarrier

Hengjun Zhao

School of Computer and Information Science
Southwest University, Chongqing, 400715, P. R. China
zhaohj2016@swu.edu.cn

1 Introduction

In our paper *Synthesizing Barrier Certificates Using Neural Networks* accepted by HSCC'20, we developed a tool named nnbarrier that can automatically learn a barrier ceritificate represented by a neural network for the safety verification of a continuous dynamical system. Here we give a short instruction to the use of nnbarrier, covering the system requirements, installation process, the structure of source codes, sample inputs, and user-defined inputs. We will emphasize what parts that were presented in the submitted paper will be covered in the instruction, for the purpose of repeatability evaluation. If there is any problem in using nnbarrier, please contact zhaohj2016@swu.edu.cn.

2 Installation

2.1 System Requirements

It is assumed that you have Python Version 3.x installed on your system. We have tested nnbarrier on Ubuntu Linux, Mac OS, and Windows (see Table 1).

2.2 Dependent Packages

It is assumed that you have the python package manager Pip installed for Python 3.x, which will facilitate the installation of dependent packages greatly.

Essentially, to run nnbarrier without visualization, the only two packages you need to install are Pytorch and NumPy. It seems that numpy will be automatically installed when installing Pytorch. Please visit https://pytorch.org/ for the installation instructions for the popular machine learning platform Pytorch. For example, with the combination Mac+Python 3.7+Pip (without cuda GPU support), the latest stable version Pytorch 1.3 can be installed by simply run

```
pip3 install torch torchvision
```

If you would like to visualize the generated barrier function together with the considered system, i.e. the system dynamics, the domain, the initial set, and the unsafe/safe region, then some additional graphics packages are required. For visualization of 2D systems, matplotlib needs to be installed, and you are referred to

https://matplotlib.org/users/installing.html for the instructions. In the implementation of nnbarrier, visulaization of 3D systems is supported by Mayavi, a 3D scientific data visualization library. Please visit http://docs.enthought.com/mayavi/mayavi/installation.html#installing-with-pip for the installation of mayavi and its dependencies (e.g. PyQt5). In our testing, on most platforms the visualization-required packages can be installed with the following commands easily:

```
pip install matplotlib
pip install mayavi
pip install PyQt5
```

where pip can be actually pip3. However, we do met some problems occasionally. If you failed to get this done in the end, nnbarrier can still be run by commenting the statements for visualization, which will be explained later.

In summary, we have tested nnbarrier using the following combinations

Table 1. Tested platforms and packages for nnbarrier

OS	Python	Pip	Pytorch	Visualization
Ubuntu 18.04.02	3.6.7	9.0.1	1.2.0	matplotlib+mayavi+PyQt5
Ubuntu 18.04.02	3.6.9	19.3.1	1.3.1	matplotlib+mayavi+PyQt5
Windows 10 1903	3.7.3	19.3.1	1.3.1	matplotlib+mayavi+PyQt5
Mac OS 10.11.6	3.7.6	19.3.1	1.3.1	matplotlib+mayavi+PySide2

2.3 Obtain the nnbarrier Package

Suppose that you have Git install on your system. Then the nnbarrier package can be obtained via

```
git clone https://github.com/zhaohj2017/HSCC20-Repeatability
```

It consists of 10 Python source files and one file folder as listed below:

- acti.py:
- ann.py:
- data.py:
- loss.py:
- lrate.py:
- main.py:
- opt.py:
- plot.py:
- plot3d.py:
- train.py
- cases: a file folder consisting of all the problem definitions in our paper

3 Sample Input

comment some lines

```
import torch.nn as nn
import numpy as np
import superp
import prob
```

```
if i==0:
    abc
else:
    def
```

python3: –version `__init__`

4 Cases in the Paper

- What elements of the paper are included in the REP (e.g.: specific figures, tables, etc.).
- Instructions for installing and running the software and extracting the corresponding results.

5 Define Your Own Problem

6 Fine-Tuning

6.1 A Subsection Sample

Please [2] try [1] avoid rasterized images for line-art diagrams and schemas. Whenever possible, use vector graphics instead (see Fig. 1).

References

1. Barry, A., Majumdar, A., Tedrake, R.: Safety verification of reactive controllers for uav flight in cluttered environments using barrier certificates. In: 2012 IEEE International Conference on Robotics and Automation, ICRA 2012. pp. 484–490. nstitute of Electrical and Electronics Engineers Inc. (2012)
2. Prajna, S., Jadbabaie, A.: Safety verification of hybrid systems using barrier certificates. In: Proceedings of the 7th International Workshop on Hybrid Systems: Computation and Control HSCC. pp. 477–492 (2004)

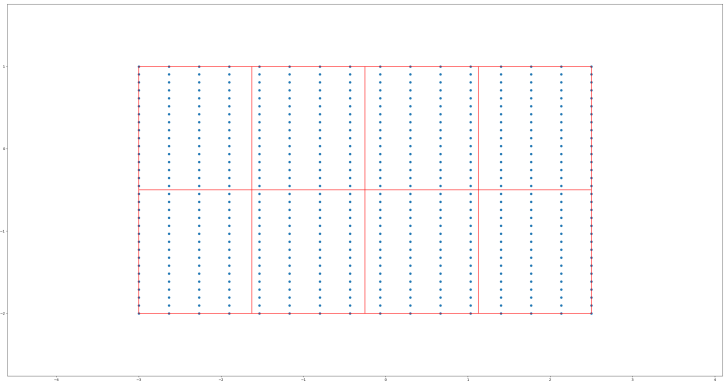

Fig. 1. A figure caption is always placed below the illustration. Please note that short captions are centered, while long ones are justified by the macro package automatically.