

Verification of Deep Convolutional Neural Network using ImageStars

The experimental results of the paper consist of three main parts

1. Comparison of the Zonotope, Polytope and ImageStar methods on three MNIST networks.
2. Comparison of the Polytope and ImageStar methods on VGG16 and VGG19.
3. Comparison of the exact and approximate scheme of the ImageStar methods on VGG16 and VGG19.

Requirements for reproducing the results

1. Matlab 2019b
2. A Computer with > 60 GiB RAM. **Note that a computer with smaller RAM cannot be used to reproduce all the results in the paper.**

Install NNV

1. Clone NNV: [git clone https://github.com/verivital/nnv](https://github.com/verivital/nnv)
2. Install NNV:
 - Open Matlab
 - Go to code/nnv/
 - Run [install.m](#)

Reproduce Part I

1. Produce Figure 8.
 - Go to [code/nnv/example/Submission/CAV2020_ImageStar/MNIST_NETS/Small](#)
 - Run [plot_ranges.m](#)
2. Produce Table 1.
 - Go to [code/nnv/example/Submission/CAV2020_ImageStar/MNIST_NETS/Small](#)
 - Run [compare_star_absdom.m](#)
3. Produce Table 2.
 - Go to [code/nnv/example/Submission/CAV2020_ImageStar/MNIST_NETS/Medium](#)
 - Run [compare_star_absdom.m](#)
4. Produce Table 3.
 - Go to [code/nnv/example/Submission/CAV2020_ImageStar/MNIST_NETS/Large](#)
 - Run [compare_star_absdom.m](#)
5. Produce Figure 13 in Appendix
 - Go to [code/nnv/example/Submission/CAV2020_ImageStar/MNIST_NETS/Architecture](#)
 - Run [plot_network_architectures.m](#)

Note for reproducing Part I:

When producing Table 1, 2, and 3, in the case that the reviewer run into “out of memory” problem, we suggest the reviewer to run the short version of the results by running

[“compare_star_absdom_short.m”](#) for each table. This script will produce a small version of the full result.

Reproduce Part II

1. Produce Table 4, VGG16 part
 - Go to [code/nnv/example/Submission/CAV2020_ImageStar/VGG16/Compare_Polytope_ImageStar/](#)
 - Run [verify_VGG16.m](#)
2. Produce Table 4, VGG19 part
 - Go to [code/nnv/example/Submission/CAV2020_ImageStar/VGG19/Compare_Polytope_ImageStar/](#)
 - Run [verify_VGG19.m](#)

Reproduce Part III

1. Produce Table 5, VGG16 part
 - Go to [code/nnv/example/Submission/CAV2020_ImageStar/VGG16/Compare_Exact_vs_Approx](#)
 - Run [verify_robustness_delta_e_07.m](#) and [verify_robustness_delta_2e_07.m](#)
2. Produce Table 5, VGG19 part
 - Go to [code/nnv/example/Submission/CAV2020_ImageStar/VGG19/Compare_Exact_vs_Approx](#)
 - Run [verify_robustness_delta_e_07.m](#) and [verify_robustness_delta_2e_07.m](#)
3. Produce Figure 9
 - Go to [code/nnv/example/Submission/CAV2020_ImageStar/VGG19/Plot_Figures](#)
 - Run [plot_vgg19_exact_range.m](#)
4. Produce Figure 10
 - Go to [code/nnv/example/Submission/CAV2020_ImageStar/VGG19/Plot_Figures](#)
 - Run [plot_vgg19_counter_example.m](#)
5. Produce Figure 11
 - Go to [code/nnv/example/Submission/CAV2020_ImageStar/VGG19/Plot_Figures](#)
 - Run [plot_vgg19_reachTime.m](#)
6. Produce Figure 12
 - Go to [code/nnv/example/Submission/CAV2020_ImageStar/VGG19/Plot_Figures](#)
 - Run [plot_vgg19_inputSize_effect.m](#)