For running the code, you need to just run “abstract.py” file that automatically run the “encoding.py” file.

Just change the path for the read functions in the code.

“abstract.py” code abstracts the neural network and gives INN, and ‘encoding.py’ code encodes INN and uses MILP solver to solve the constraints. Also, the inputs are given in the “experiments” folder.

Matlab files consider different combinations of merged neurons for ‘Converted_ACASXU_run2a_1_7_batch_2000’ NN.

First run C2.m, C4.m, … then params.m, then for drawing the abstraction time and other figures, ‘different_timing.m’ code is used and also ‘output.m’ code is used for drawing the output range.

You can use https://matlab.mathworks.com/ for using matlab online.

Inside experiments folder:

Input_coeficient folder shows the coefficient for equations to create a convex hall as an input of INN.

INN_Rluplex shows the weights and biases of different three INNs because first we need to convert ACAS NN to our format by the code “Format_changing_reluplex2us.py”.

Random composition shows the number of different merged neurons in each layer and inside of each of folders there is a code to create different permutations for a fixed number of neurons in each layer randomly.