```
Last login: Thu Jul 22 14:32:56 on ttys001
exec '/Applications/Julia-1.6.app/Contents/Resources/julia/bin/julia'
Chioma_N@Akwarandus-MacBook-Air ~ % exec '/Applications/Julia-1.6.app/Contents/R
esources/julia/bin/julia'

               _
   _       _ _(_)_     |  Documentation: https://docs.julialang.org
  (_)     | (_) (_)    |
   _ _   _| |_  __ _   |  Type "?" for help, "]?" for Pkg help.
  | | | | | | |/ _` |  |
  | | |_| | | | (_| |  |  Version 1.6.1 (2021-04-23)
 _/ |\__'_|_|_|\__'_|  |  Official https://julialang.org/ release
|__/                   |

julia> using NeuralVerification, LazySets

julia> import NeuralVerification: ReLU, Id

julia> lowerInput = [1.0, 3.0, 0.0,
       5.0, 7.0, 0.0,
       9.0, 11.0, 0.0,
       13.0, 15.0, 0.0,
       17.0, 19.0, 0.0,
       21.0, 23.0, 0.0,
       25.0, 27.0, 0.0,
       29.0, 31.0, 0.0,
       34.0, 36.0, 0.0,
       38.0, 40.0, 0.0]
30-element Vector{Float64}:
  1.0
  3.0
  0.0
  5.0
  7.0
  0.0
  9.0
 11.0
  0.0
 13.0
  ⋮
 29.0
 31.0
  0.0
 34.0
 36.0
  0.0
 38.0
 40.0
  0.0

julia> upperInput = [4.0, 6.0, 5.0,
       8.0, 10.0, 5.0,
       12.0, 14.0, 5.0,
```

```
       16.0, 18.0, 5.0,
       20.0, 22.0, 5.0,
       24.0, 26.0, 5.0,
       28.0, 30.0, 5.0,
       32.0, 34.0, 5.0,
       37.0, 39.0, 5.0,
       41.0, 43.0, 5.0]
30-element Vector{Float64}:
  4.0
  6.0
  5.0
  8.0
 10.0
  5.0
 12.0
 14.0
  5.0
 16.0
 18.0
  5.0
 20.0
 22.0
  5.0
 24.0
 26.0
  5.0
 28.0
 30.0
  5.0
 32.0
 34.0
  5.0
 37.0
 39.0
  5.0
 41.0
 43.0
  5.0

julia> x = Hyperrectangle(low = lowerInput, high = upperInput)
Hyperrectangle{Float64, Vector{Float64}, Vector{Float64}}([2.5, 4.5, 2.5, 6.5, 8
.5, 2.5, 10.5, 12.5, 2.5, 14.5  …  2.5, 30.5, 32.5, 2.5, 35.5, 37.5, 2.5, 39.5,
41.5, 2.5], [1.5, 1.5, 2.5, 1.5, 1.5, 2.5, 1.5, 1.5, 2.5, 1.5  …  2.5, 1.5, 1.5,
 2.5, 1.5, 1.5, 2.5, 1.5, 1.5, 2.5])

julia> y = Hyperectangle(low = [-10000], high = [-0.1])
ERROR: UndefVarError: Hyperectangle not defined
Stacktrace:
 [1] top-level scope
   @ REPL[6]:1

julia> y = Hyperrectangle(low = [-10000], high = [-0.1])
```

```
Hyperrectangle{Float64, Vector{Float64}, Vector{Float64}}([-5000.05], [4999.95])

julia> model = "/Users/Chioma_N/Desktop/object/PCC-RL-master/src/gym/tensorflow/
reLuNet/resNet"
"/Users/Chioma_N/Desktop/object/PCC-RL-master/src/gym/tensorflow/reLuNet/resNet"

julia> net = read_nnet(model)
Network(NeuralVerification.Layer[NeuralVerification.Layer{ReLU, Float64}([-0.476
940215 -0.233635619 … 0.260245979 0.0854543895; -0.279218972 0.212435156 … -0.30
4853797 -0.287973344; … ; 0.043456167 -0.299254805 … -0.336141109 0.0869233087;
0.0609097183 -0.0746256411 … -0.356662869 -0.211450607], [-0.03726024, 0.0, 0.0,
 0.06048633, -0.11259795, 0.0, 0.0, -0.07211282, 0.044972, 0.0, 0.0, 0.0, 0.0, 0
.01267096, -0.00579271, 0.0], ReLU()), NeuralVerification.Layer{ReLU, Float64}([
-0.02278417 -0.15391123 … -0.40684664 -0.21582426; 0.27223375 0.23778197 … 0.137
33128 0.12970361; … ; 0.18829295 0.33743098 … 0.06836078 0.16406111; 0.37219688
-0.16067436 … -0.24720219 0.12515154], [0.0, 0.0, 0.06707639, 0.0, -0.07552656,
0.00930779, -0.00740244, 0.0, -0.04904214, 0.0, -0.02598077, 0.03745193, 0.03417
419, -0.01520709, 0.0, 0.01063646], ReLU()), NeuralVerification.Layer{Id, Float6
4}([0.30263627 -0.23838618 … -0.09799442 -0.00773645], [0.04735997], Id())])

julia> prob = Problem(net, x, y)
Problem{Hyperrectangle{Float64, Vector{Float64}, Vector{Float64}}, Hyperrectangl
e{Float64, Vector{Float64}, Vector{Float64}}}(Network(NeuralVerification.Layer[N
euralVerification.Layer{ReLU, Float64}([-0.476940215 -0.233635619 … 0.260245979
0.0854543895; -0.279218972 0.212435156 … -0.304853797 -0.287973344; … ; 0.043456
167 -0.299254805 … -0.336141109 0.0869233087; 0.0609097183 -0.0746256411 … -0.35
6662869 -0.211450607], [-0.03726024, 0.0, 0.0, 0.06048633, -0.11259795, 0.0, 0.0
, -0.07211282, 0.044972, 0.0, 0.0, 0.0, 0.0, 0.01267096, -0.00579271, 0.0], ReLU
()), NeuralVerification.Layer{ReLU, Float64}([-0.02278417 -0.15391123 … -0.40684
664 -0.21582426; 0.27223375 0.23778197 … 0.13733128 0.12970361; … ; 0.18829295 0
.33743098 … 0.06836078 0.16406111; 0.37219688 -0.16067436 … -0.24720219 0.125151
54], [0.0, 0.0, 0.06707639, 0.0, -0.07552656, 0.00930779, -0.00740244, 0.0, -0.0
4904214, 0.0, -0.02598077, 0.03745193, 0.03417419, -0.01520709, 0.0, 0.01063646]
, ReLU()), NeuralVerification.Layer{Id, Float64}([0.30263627 -0.23838618 … -0.09
799442 -0.00773645], [0.04735997], Id())]), Hyperrectangle{Float64, Vector{Float
64}, Vector{Float64}}([2.5, 4.5, 2.5, 6.5, 8.5, 2.5, 10.5, 12.5, 2.5, 14.5  …  2
.5, 30.5, 32.5, 2.5, 35.5, 37.5, 2.5, 39.5, 41.5, 2.5], [1.5, 1.5, 2.5, 1.5, 1.5
, 2.5, 1.5, 1.5, 2.5, 1.5  …  2.5, 1.5, 1.5, 2.5, 1.5, 1.5, 2.5, 1.5, 1.5, 2.5])
, Hyperrectangle{Float64, Vector{Float64}, Vector{Float64}}([-5000.05], [4999.95
]))

julia> res = solve(ReluVal(max_iter=100), prob)
CounterExampleResult(:violated, [2.5, 4.5, 2.5, 6.5, 8.5, 2.5, 10.5, 12.5, 2.5,
14.5  …  2.5, 30.5, 32.5, 2.5, 35.5, 37.5, 2.5, 39.5, 41.5, 2.5])

julia> NeuralVerification.compute_output(net, res.counter_example)
1-element Vector{Float64}:
 1.0379898681228816

julia> res.counter_example
30-element Vector{Float64}:
  2.5
```

```
  4.5
  2.5
  6.5
  8.5
  2.5
 10.5
 12.5
  2.5
 14.5
 16.5
  2.5
 18.5
 20.5
  2.5
 22.5
 24.5
  2.5
 26.5
 28.5
  2.5
 30.5
 32.5
  2.5
 35.5
 37.5
  2.5
 39.5
 41.5
  2.5

julia>
```