ROBUSTNESS VERIFIER HEURITICS FOR NEURAL NETWORKS

R. Delialisi

ETH Zürich, D-INFK

C. Trassoudaine[*]

IMT Atlantique,
EURECOM, Data Science dpt.

1. PROBLEM DEFINITION AND ANALYSIS TECHNIQUES

The purpose of this work is to prove formally the local robustness (from now referred as robustness) with respect to some inputs of neural-networks (N.N.) by heuristically combining box analysis (B.A.) and linear-programming (L.-P.) solving. In this paper, we aim to present a time-efficient way of verifying N.N. robustness for large ϵ-perturbations, using the L_∞ norm.

Box analysis A very simple and fast approach to solve the N.N. robustness problem is to use a polyhedra abstract domain and to propagate it across the layers.

Linear programming A more precise but more computationally intensive technique to verify the robustness of a N.N. is to use a linear solver to propagate the box while approximating better the non-linearities. This can be used of two ways: compute the bounds for a given layer using modeling constraints and the previous layers bounds, or directly check for robustness on the last layers based on the approximated previous layers bounds.

We generaly prefer using L.-P. on the first layers of the N.N. not to propagate growing boxes. An analogy can be drawn with low-noise amplifiers in the physics field, for which it is really important to keep the noise minimal at the beginning of the process.

2. HEURISTICS

In this section we will refer to Box Analysis as ELINA and Linear Programmming range propagation as GUROBI within the algorithm sections as these are the used tools in our implementation.

Greedy approach The first approach and most efficient one for small N.N.s is to iterate over the layers and using L.-P. at each step and verifying with B.A. as an exit condition. This takes advantage of the cheap computational cost of B.A. and of the incremental precision of L.-P., which has a cost. The biggest verifiable ϵ-perturbations will take more time than the smaller ones due to the greater number of L.-P. calls, but this also returns results very fast for small perturbations.

Algorithm 1 Greedy precise algorithm with an early stopping condition

$n = size(NN)$
for $i \in [0, n]$ **do**
  GUROBI_OPTIMIZE($Layer_{i+1}$)
  if $Layer_{i+1}$ is the last layer **then**
    return VERIFY($Layer_n$)
  else
    ELINA_RANGE($Layer_{i+1,\ldots,n}$)
    if VERIFY($Layer_n$) **then**
      return true
    end if
  end if
end for

Large networks strategy For networks with layers of more than 500 neurons the above strategy is too slow. The strategy we follow is to optimize as many neurons as we can in the earlier layers using L.-P., optimize the rest of the net with B.A. and do a final L.-P. on the output.

Algorithm 2 Large networks heuristic

$n = size(NN)$
/* **Compute boxes using ELINA.** */
$(Range_1, \ldots, Range_n)$ = ELINA_RANGE($Layer_{1,\ldots,n}$)
if VERIFY($Layer_n$) **then**
  return true
else
  /* **Compute importance of the neurons as the sum of the absolute values of the outgoing weights times the upper bound.** */
  w = COMPUTE_WEIGHTS($Range_n$)
  /* **Optimize most weighted neurons on the second hidden layer based on importance values.** */
  $opt_{neurons}$ = index(w, threshold)
  $Range_{2,opt_{neurons}}$ = GUROBI_OPTIMIZE($Layer_{2,opt_{neurons}}$)
  /* **Compute remaining boxes using ELINA.** */
  $(Range_3, \ldots, Range_n)$ = ELINA_RANGE($Layer_{3,\ldots,n}$)
  return VERIFY($Layer_n$)
end if

In the following, intervals range from green (smallests) to red.

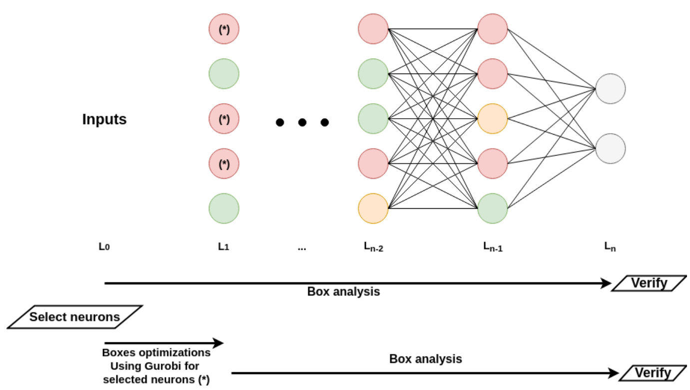

Fig. 1. Large networks strategy

3. PERFORMANCES

The most important factors in the L.-P. analysis complexity seems to be the number of neurons per layer as well as the magnitude of the perturbation. We determined experimentally a $o(exp(\epsilon))$ complexity to certify a given N.N.. Hence, reducing the range amplitude for the first layers by using L.-P. also help to reduce the computational time.

Layers	neurons per layers
3	10, 20, 50
4	1024
6	20, 30, 100, 200
9	100, 200

Fig. 2. Tested configurations

The model performs the analysis in a few minutes for all of the networks and ϵ-perturbations ranging from 10^{-2} to 10^{-4} using the greedy algorithm for deep-N.N.s and the Large-net strategy for the narrow-N.N.s.

*. Work performed while at ETH Zürich