Boosting Robustness Certification of Neural Networks

Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev
Department of Computer Science, ETH Zurich

Problem: Certification of neural network robustness

Small changes in pixel intensities can cause neural networks to misclassify images. An $\epsilon$-norm based perturbation $x'$ of an input $x$ can be written as $x' = x + \epsilon \cdot s$, where $s$ is a unit vector (considering only one dimension). If a neural network $f$ classifies $x$ as $y$, then it should classify $x'$ as $y$. This is formalized in the following equation:

$$\|x' - y\|_\infty = \|x + \epsilon \cdot s - y\|_\infty = \|\epsilon \cdot s\|_\infty = \epsilon$$

The neural network classifies the image $x'$ correctly as $b$ if it is within a radius $\epsilon$.

Goal: certify if a given neural network correctly classifies all images $x$ in the test set $X$, i.e., all images $x$ where the intensity of each pixel in $x$ differs by at most $\epsilon$ from the corresponding pixel in $x'$. 

Solving the above problem yields the output affine form in $\epsilon$-ball around the input $x$. It is an overapproximation of the output of the Neural Network Function $f$.

Key Idea: Best of both worlds

Solvers + Abstract Interpretation

Deep on a toy feedforward network with ReLU

Faster (most efficient) or more precise (less efficient) solvers are selected for refinement.

End to end implementation

- Anytime relaxation
  - refine $f$ iteration of neurons in a layer with a timeout $T$ for the solver
  - refine $f(x)$ for a neuron in a layer with timeout $T$ for the solver
- Neuron selection heuristic for refining
  - Neurons are sorted by width and the sum of absolute output weights
  - ranks of neurons in both orders are added as candidates for refinement

Refinement bounds

Revised bounds are the refined bounds. The slope of the two non-vertical parallel blue lines is $\lambda = u_0/(u_0 - u_0')$, and the slope of the two non-vertical parallel green lines is $\lambda' = u_0/(u_0 - u_0')$. The blue parallelogram is for computing the output affine form in DeepZ, whereas the green parallelogram is for computing the output of the refined ReLU transformer considered in this work.

RefineZono: Our system for neural network robustness

Our approach on the toy network

End to end implementation

Complete verification

MNIST 3 \times 50 Network

Certify with DeepZ first, if it fails then formulate certification as MILP using per-neuron bounds produced by DeepZ.

Results with RefineZono: State-of-the-art precision and scalability

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST</td>
<td>5 \times 100</td>
<td>0.07</td>
<td>0.03</td>
</tr>
<tr>
<td></td>
<td>5 \times 100</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>9 \times 100</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>5 \times 100</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>5 \times 100</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>9 \times 100</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>ACAS Xu</td>
<td>6 \times 100</td>
<td>0.0012</td>
<td>0.0012</td>
</tr>
<tr>
<td></td>
<td>6 \times 100</td>
<td>0.03</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Evaluation

- 3 \times 50 FNN and all CNNs on a 2.6 GHz 14 core Intel Xeon CPU E5-2690
- All remaining FNNs on a 3.3 GHz 10 core Intel X9700 Skylake CPU

Benchmarks

- property 9 defined in [5] for the ACAS Xu network
- correctly classified images among the first 100 test images for the rest

References

[3] Differentiable Neural Interpreters for Provably Robust Neural Networks: ICML'18

Further reading on efficient abstract interpretation: elinai.ethz.ch