Piecewise Linear Neural Network Verification
A Comparative Study

Rudy Bunel, Ilker Turkaslan, Philip HS Torr, Pushmeet Kohli, M Pawan Kumar
Problem

Assume that you are given:
- The weights of a Neural Network
- Some property that should hold over a certain region of the input domain

You want to prove that the property always holds.

Example applications:
- If I’m less than \(d\) away from a point, can I guarantee the same prediction?
- If inputs are in range, can I guarantee the predicted class is \(c\) ?
- Is the numerical value of output \(o\) always below limit value \(l\) ?
Formalism

For a Neural Network that implements the function
\[ y = f(x) \]

We want to prove that:
\[ x \in C \]
\[ y = f(x) \]
\[ \Rightarrow \]
\[ P(\ y \ ) \]
Example – Robustness to Adversarial Examples

\[ \| x - 1 \|_\infty \leq \varepsilon \]

\[ x \in C \]
\[ y = f(x) \]
\[ \Rightarrow \]
\[ P(y) \]

\[
\begin{pmatrix}
1 \\
-1 \\
0 \\
\vdots \\
0
\end{pmatrix} \geq 0
\]
Example – Bound on output

\[ x \in C \]
\[ y = f(x) \]
\[ \Rightarrow \]
\[ P(y) \]

Prove that \( y > -5 \)
What about more complex properties?

OR clauses:

\[ P(y) \triangleq [y_1 \geq 0] \lor [y_2 \geq 0] \lor [y_3 \geq 0] \iff \max(y_1, y_2, y_3) \geq 0 \]

Can be implemented by adding Maxpooling layer at the end
What about more complex properties?

AND clauses:

\[ P(y) \triangleq [y_1 \geq 0] \land [y_2 \geq 0] \land [y_3 \geq 0] \]

\[ \iff \]

\[ \min(y_1, y_2, y_3) \geq 0 \]

\[ \iff \]

\[ -\max(-y_1, -y_2, -y_3) \geq 0 \]

\[ \text{Can be implemented by adding Linear + Maxpooling + Linear} \]
Canonical problem

For any property that is a Boolean formula over linear inequalities,

We can just build an equivalent network for which we prove a bound
Example – Bound on output

Prove that $y > -5$
Strategy: counterexample search

Create a feasibility problem for the existence of a counterexample.

If the problem is feasible:
   then a counterexample is shown, the property is False

If the problem is infeasible:
   then no counterexample can exist, the property is True
Counter example search

Prove that $y > -5$

$-2 \leq x_1 \leq 2$
$-2 \leq x_2 \leq 2$
$a_{in} = x_1 + x_2$
$b_{in} = x_1 - x_2$
$a_{out} = \max(a_{in}, 0)$
$b_{out} = \max(b_{in}, 0)$
$y = -a_{out} - b_{out}$
$y \leq -5$
Difficulty

\[-2 \leq x_1 \leq 2\]
\[-2 \leq x_2 \leq 2\]
\[a_{in} = x_1 + x_2\]
\[b_{in} = x_1 - x_2\]
\[a_{out} = \max(a_{in}, 0)\]
\[b_{out} = \max(b_{in}, 0)\]
\[y = -a_{out} - b_{out}\]
\[y \leq -5\]

\[a_{out} = \max(a_{in}, 0)\]
\[b_{out} = \max(b_{in}, 0)\]

The problem is in the non-linearities!
Strategy 1: MIP solver

In Gurobi we trust


Strategy 1: Mixed integer encoding

\[ a_{out} = \max(a_{in}, 0) \]

\[ a_{out} \geq a_{in} \]
\[ a_{out} \geq 0 \]

\[ a_{out} \leq a_{in} + (1 - \delta_a)M_a \]
\[ a_{out} \leq \delta_a M_a \]

\[ \delta_a \in \{0, 1\} \]

where \( M_a \) is big enough
Strategy 1: Mixed integer encoding

If $\delta_a = 0$

$$0 \leq a_{out} \leq 0$$

If $\delta_a = 1$

$$a_{in} \leq a_{out} \leq a_{in}$$

$$a_{out} \geq a_{in}$$

$$a_{out} \geq 0$$

$$a_{out} \leq a_{in} + (1 - \delta_a)M_a$$

$$a_{out} \leq \delta_a M_a$$

$$\delta_a \in \{0, 1\}$$

Problem: Mixed Integer Programming is also hard.
Strategy 2: Reluplex

Modify assignments until all constraints are satisfied.
If you feel stuck, divide the problem into two easier ones

Strategy 2: Reluplex

$$-2 \leq x_1 \leq 2$$
$$-2 \leq x_2 \leq 2$$
$$a_{in} = x_1 + x_2$$
$$b_{in} = x_1 - x_2$$
$$a_{out} = \max(a_{in}, 0)$$
$$b_{out} = \max(b_{in}, 0)$$
$$y = -a_{out} - b_{out}$$
$$y \leq -5$$

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$a_{in}$</th>
<th>$a_{out}$</th>
<th>$b_{in}$</th>
<th>$b_{out}$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Fix Linear constraints
Strategy 2: Reluplex

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$a_{in}$</th>
<th>$a_{out}$</th>
<th>$b_{in}$</th>
<th>$b_{out}$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>-5</td>
</tr>
</tbody>
</table>

Fix one non-linear constraint (b)

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$a_{in}$</th>
<th>$a_{out}$</th>
<th>$b_{in}$</th>
<th>$b_{out}$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>-5</td>
</tr>
</tbody>
</table>

$-2 \leq x_1 \leq 2$
$-2 \leq x_2 \leq 2$
$a_{in} = x_1 + x_2$
$b_{in} = x_1 - x_2$
$a_{out} = \max(a_{in}, 0)$
$b_{out} = \max(b_{in}, 0)$
$y = -a_{out} - b_{out}$
$y \leq -5$
Strategy 2: Reluplex

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$a_{in}$</th>
<th>$a_{out}$</th>
<th>$b_{in}$</th>
<th>$b_{out}$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>-5</td>
</tr>
<tr>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fix Linear constraints</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-2</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>-5</td>
</tr>
</tbody>
</table>

- $-2 \leq x_1 \leq 2$
- $-2 \leq x_2 \leq 2$
- $a_{in} = x_1 + x_2$
- $b_{in} = x_1 - x_2$
- $a_{out} = \max(a_{in}, 0)$
- $b_{out} = \max(b_{in}, 0)$
- $y = -a_{out} - b_{out}$
- $y \leq -5$
Strategy 2: Reluplex

If no progress is made (the same non-linearity is fixed too many times), the problem is split into two and the search continues.

\[-2 \leq x_1 \leq 2\]
\[-2 \leq x_2 \leq 2\]
\[a_{in} = x_1 + x_2\]
\[b_{in} = x_1 - x_2\]
\[a_{out} = \max(a_{in}, 0)\]
\[b_{out} = \max(b_{in}, 0)\]
\[y = -a_{out} - b_{out}\]
\[y \leq -5\]
Strategy 3: Planet

Divide the problem until you reach infeasibilities, then backtrack

Ruediger Ehlers. 
*Formal verification of piece-wise linear feed-forward neural networks.* 
Automated Technology for Verification and Analysis, 2017
Strategy 3: Planet

Building a linear approximation of the network:

→ Replace non-linearities by set of constraints
Strategy 3: Planet

\[-2 \leq x_1 \leq 2 \]
\[-2 \leq x_2 \leq 2 \]
\[a_{in} = x_1 + x_2 \]
\[b_{in} = x_1 - x_2 \]
\[a_{out} = \max(a_{in}, 0) \]
\[b_{out} = \max(b_{in}, 0) \]
\[y = -a_{out} - b_{out} \]
\[y \leq -5 \]
Strategy 3: Planet

\[ a_{out} = 0 \]
\[ a_{out} = a_{in} \]

\[ b_{out} = 0 \]
\[ b_{out} = b_{in} \]
\[ b_{out} = 0 \]
\[ b_{out} = b_{in} \]

\[-2 \leq x_1 \leq 2\]
\[-2 \leq x_2 \leq 2\]
\[ a_{in} = x_1 + x_2\]
\[ b_{in} = x_1 - x_2\]

\[ a_{out} \geq 0, \quad a_{out} \geq a_{in}\]
\[ a_{out} \leq \frac{a_{in}}{2} + 2\]

\[ b_{out} \geq 0, \quad b_{out} \geq b_{in}\]
\[ b_{out} \leq \frac{b_{in}}{2} + 2\]

\[ y = -a_{out} - b_{out}\]
\[ y \leq -5\]
Strategy 3: Planet

Is it feasible?

No

\[-2 \leq x_1 \leq 2\]
\[-2 \leq x_2 \leq 2\]
\[a_{in} = x_1 + x_2\]
\[b_{in} = x_1 - x_2\]
\[a_{out} \geq 0, \quad a_{out} \geq a_{in}\]
\[a_{out} \leq \frac{a_{in}}{2} + 2\]
\[b_{out} \geq 0, \quad b_{out} \geq b_{in}\]
\[b_{out} \leq \frac{b_{in}}{2} + 2\]
\[y = -a_{out} - b_{out}\]
\[y \leq -5\]
\[a_{out} = 0\]
Strategy 3: Planet

Is it feasible?
Yes \((x_1 = 2, x_2 = 2, a_{\text{out}} = 4, b_{\text{out}} = 2)\)

\(-2 \leq x_1 \leq 2\)
\(-2 \leq x_2 \leq 2\)
\(a_{\text{in}} = x_1 + x_2\)
\(b_{\text{in}} = x_1 - x_2\)
\(a_{\text{out}} \geq 0, \quad a_{\text{out}} \geq a_{\text{in}}\)
\(a_{\text{out}} \leq \frac{a_{\text{in}}}{2} + 2\)
\(b_{\text{out}} \geq 0, \quad b_{\text{out}} \geq b_{\text{in}}\)
\(b_{\text{out}} \leq \frac{b_{\text{in}}}{2} + 2\)
\(y = -a_{\text{out}} - b_{\text{out}}\)
\(y \leq -5\)
\(a_{\text{out}} = a_{\text{in}}\)
Strategy 3: Planet

\[ a_{\text{out}} = 0 \]

\[ a_{\text{out}} = a_{\text{in}} \]

\[ b_{\text{out}} = 0 \]
\[ b_{\text{out}} = b_{\text{in}} \]

\[ b_{\text{out}} = 0 \]
\[ b_{\text{out}} = b_{\text{in}} \]

Is it feasible?

No

\[ -2 \leq x_1 \leq 2 \]
\[ -2 \leq x_2 \leq 2 \]
\[ a_{\text{in}} = x_1 + x_2 \]
\[ b_{\text{in}} = -x_1 - x_2 \]
\[ a_{\text{out}} \geq 0, \quad a_{\text{out}} \geq a_{\text{in}} \]
\[ a_{\text{out}} \leq \frac{a_{\text{in}}}{2} + 2 \]
\[ b_{\text{out}} \geq 0, \quad b_{\text{out}} \geq b_{\text{in}} \]
\[ b_{\text{out}} \leq \frac{b_{\text{in}}}{2} + 2 \]
\[ y = -a_{\text{out}} - b_{\text{out}} \]
\[ y \leq -5 \]
\[ a_{\text{out}} = a_{\text{in}} \]
\[ b_{\text{out}} = 0 \]
Strategy 3: Planet

Is it feasible?
No

\[-2 \leq x_1 \leq 2\]
\[-2 \leq x_2 \leq 2\]
\[a_{\text{in}} = x_1 + x_2\]
\[b_{\text{in}} = -x_1 - x_2\]
\[a_{\text{out}} \geq 0, \quad a_{\text{out}} \geq a_{\text{in}}\]
\[a_{\text{out}} \leq \frac{a_{\text{in}}}{2} + 2\]
\[b_{\text{out}} \geq 0, \quad b_{\text{out}} \geq b_{\text{in}}\]
\[b_{\text{out}} \leq \frac{b_{\text{in}}}{2} + 2\]
\[y = -a_{\text{out}} - b_{\text{out}}\]
\[y \leq -5\]
\[a_{\text{out}} = a_{\text{in}}\]
\[b_{\text{out}} = b_{\text{in}}\]
Strategy 3: Planet

There is no feasible assignment
No counterexample can exist.
The property is True

\[-2 \leq x_1 \leq 2\]
\[-2 \leq x_2 \leq 2\]

\[a_{in} = x_1 + x_2\]
\[b_{in} = -x_1 - x_2\]

\[a_{out} \geq 0, \quad a_{out} \geq a_{in}\]
\[a_{out} \leq \frac{a_{in}}{2} + 2\]

\[b_{out} \geq 0, \quad b_{out} \geq b_{in}\]
\[b_{out} \leq \frac{b_{in}}{2} + 2\]

\[y = -a_{out} - b_{out}\]
\[y \leq -5\]
Strategy 4: Branch and Bound

Evaluate worst case

If it’s too bad, split the problem into smaller ones
Strategy 4: Branch and Bound

Use Planet approximation, and compute the minimum of $y$

\[-2 \leq x_1 \leq 2\]
\[-2 \leq x_2 \leq 2\]
\[a_{in} = x_1 + x_2\]
\[b_{in} = x_1 - x_2\]
\[a_{out} \geq 0, \quad a_{out} \geq a_{in}\]
\[a_{out} \leq \frac{a_{in}}{2} + 2\]
\[b_{out} \geq 0, \quad b_{out} \geq b_{in}\]
\[b_{out} \leq \frac{b_{in}}{2} + 2\]
\[y = -a_{out} - b_{out}\]

Minimum of $y$, subject to all the constraints:

$y_{min} = -6$

We look for $y \leq -5$
So there might be counterexample in the region
Strategy 4: Branch and Bound

Use Planet approximation, and compute the minimum of $y$

$$-2 \leq x_1 \leq 0$$
$$-2 \leq x_2 \leq 2$$
$$a_{in} = x_1 + x_2$$
$$b_{in} = -x_1 - x_2$$
$$a_{out} \geq 0, \quad a_{out} \geq a_{in}$$
$$a_{out} \leq \frac{a_{in}}{3} + \frac{4}{3}$$
$$b_{out} \geq 0, \quad b_{out} \geq b_{in}$$
$$b_{out} \leq \frac{b_{in}}{3} + \frac{4}{3}$$
$$y = -a_{out} - b_{out}$$

Minimum of $y$, subject to all the constraints:
$$y_{min} = -2.66$$

We look for $y \leq -5$
So no possible counterexample on this region
Strategy 4: Branch and Bound

Use Planet approximation, and compute the minimum of $y$

$$0 \leq x_1 \leq 2$$
$$-2 \leq x_2 \leq 2$$
$$a_{in} = x_1 + x_2$$
$$b_{in} = -x_1 - x_2$$
$$a_{out} \geq 0, \quad a_{out} \geq a_{in}$$
$$a_{out} \leq \frac{2 a_{in}}{3} + \frac{4}{3}$$
$$b_{out} \geq 0, \quad b_{out} \geq b_{in}$$
$$b_{out} \leq \frac{2 b_{in}}{3} + \frac{4}{3}$$
$$y = -a_{out} - b_{out}$$

Minimum of $y$, subject to all the constraints:
$$y_{min} = -5.33$$

We look for $y \leq -5$
So no possible counterexample on this region
Strategy 4: Branch and Bound

Use Planet approximation, and compute the minimum of $y$

\[
0 \leq x_1 \leq 2 \\
-2 \leq x_2 \leq 0 \\
a_{in} = x_1 + x_2 \\
b_{in} = -x_1 - x_2 \\
a_{out} \geq 0, \ a_{out} \geq a_{in} \\
a_{out} \leq \frac{a_{in}}{2} + 1 \\
b_{out} \geq 0, \ b_{out} \geq b_{in} \\
b_{out} \leq b_{in} \\
y = -a_{out} - b_{out}
\]

Minimum of $y$, subject to all the constraints:
Etc ...
Strategy 4: Branch and Bound

Should the approximation be rebuilt at each step?

- Much faster to reuse the existing approximation
- But the bounds don’t really improve
Experimental Showdown

Who’s the fastest of them all?
It depends on the properties \(_(ツ)_/\_~

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Success Rate</th>
<th>Average runtime</th>
<th>Number of Wins</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rate SAT</td>
<td>UNSAT</td>
<td></td>
</tr>
<tr>
<td>Reluplex</td>
<td>79.79%</td>
<td>2077.7 s</td>
<td>1868.3 s</td>
</tr>
<tr>
<td>Planet</td>
<td>46.28%</td>
<td>7200 s</td>
<td>2532.8 s</td>
</tr>
<tr>
<td>MIP</td>
<td>49.47%</td>
<td>6014.0 s</td>
<td>2708.2 s</td>
</tr>
<tr>
<td>BaB</td>
<td>83.51%</td>
<td>128.14 s</td>
<td>733.55 s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Success Rate</th>
<th>Average runtime</th>
<th>Number of Wins</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rate SAT</td>
<td>UNSAT</td>
<td></td>
</tr>
<tr>
<td>Reluplex</td>
<td>99.4 %</td>
<td>1.14 s</td>
<td>1.17 s</td>
</tr>
<tr>
<td>Planet</td>
<td>100 %</td>
<td>0.50 s</td>
<td>0.18 s</td>
</tr>
<tr>
<td>MIP</td>
<td>100 %</td>
<td>0.66 s</td>
<td>0.65 s</td>
</tr>
<tr>
<td>BaB</td>
<td>100 %</td>
<td>5.27 s</td>
<td>29.05 s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Success Rate</th>
<th>Average Runtime</th>
<th>Number of Wins</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reluplex</td>
<td>53.09%</td>
<td>2061.8s</td>
<td>1</td>
</tr>
<tr>
<td>Planet</td>
<td>62.96%</td>
<td>2878.5s</td>
<td>13</td>
</tr>
<tr>
<td>MIP</td>
<td>65.43%</td>
<td>2647.0s</td>
<td>41</td>
</tr>
<tr>
<td>BaB</td>
<td>28.40%</td>
<td>5248.1s</td>
<td>4</td>
</tr>
</tbody>
</table>

ACAS dataset

CollisionDetection

TwinStream
What next?

Branch-and-Bound is a flexible framework:

• How to choose the splitting to prioritize?

• Can we estimate a good scale to start the bounding process?

• Can we get better lower bounds?
Better bounds (in progress)

What does better bounds means?

Zico Kolter, Eric Wong. 
_Provable defenses against adversarial examples via the convex outer adversarial polytope._
Arxiv:1711.00851
Better Bounds

\[ \text{min } y \]

such that

\[ -2 \leq x_1 \leq 2 \]
\[ -2 \leq x_2 \leq 2 \]
\[ a_{in} = x_1 + x_2 \]
\[ b_{in} = x_1 - x_2 \]
\[ a_{out} \geq 0, \quad a_{out} \geq a_{in} \]
\[ a_{out} \leq \frac{a_{in}}{2} + 2 \]
\[ b_{out} \geq 0, \quad b_{out} \geq b_{in} \]
\[ b_{out} \leq \frac{b_{in}}{2} + 2 \]
\[ y = -a_{out} - b_{out} \]

Primal problem

\[ \text{max } -2(\mu_{0,b} + \mu_{0,a}) - 2(\epsilon_{u1} + \epsilon_{u2}) \]
\[ -2(\epsilon_{l1} + \epsilon_{l2}) \]

such that

\[ \mu_{0,a} = 1 + \rho_{0,a} + \tau_{0,a} \]
\[ \mu_{0,b} = 1 + \rho_{0,b} + \tau_{0,b} \]
\[ \mu_{0,a} = 2 \left( \tau_{0,a} + \lambda_{0,a} \right) \]
\[ \mu_{0,b} = 2 \left( \tau_{0,b} + \lambda_{0,b} \right) \]
\[ \lambda_{0,a} + \lambda_{0,b} = \epsilon_{u1} - \epsilon_{l1} \]
\[ \lambda_{0,a} - \lambda_{0,b} = \epsilon_{u2} - \epsilon_{l2} \]
\[ \varepsilon \geq 0, \quad \mu \geq 0, \]
\[ \rho \geq 0, \quad \tau \geq 0 \]

Dual problem
Better Bounds

We know that optimal of primal problem is the same as the optimal of the dual problem.

(Strong Duality)

But we also know that any feasible solution to the dual problem is a valid lower bound on the optimal solution of the primal problem

(Weak Duality)
Better Bounds

\[
\begin{align*}
\text{Global minimum of the Neural Network} & \geq \\ 
\text{Minimum (unique) of the NN linear approximation} & = \\ 
\text{Maximum (unique) of the dual of the NN linear approximation} & \geq \\ 
\text{Any feasible point of the dual of the NN linear approximation} & 
\end{align*}
\]

Don’t need to solve the dual problem exactly! Any heuristic will work!

How well does it perform?
Approximate Dual solution for BaB

The bound computed are worse

But they are significantly faster to compute!
Collaborators

Pushmeet Kohli  
*Deepmind*

Philip Torr  
*University of Oxford*

Pawan Kumar  
*University of Oxford*
*Alan Turing Institute*

Ilker Turkaslan  
*University of Oxford*
Thank you for your attention. Any questions?

rudy@robots.ox.ac.uk