Incremental Bounded Model Checking of Artificial Neural Networks in CUDA

Luiz H. Sena, Iury V Bessa, Lucas C. Cordeiro, Mikhail R. Gadelha and Edjard Mota
Summary

• Introduction
• Preliminaries
• Incremental BMC of ANNs in CUDA
  • Verification of Covering Methods
  • Verification of Adversarial Cases
• Experimental Evaluation
• Conclusion
AI in Safety Critical Systems

- Marking regions to examine;
  - Red quadrilateral is marked by ANN and blue quadrilateral is marked by the radiologist.

- Mass detection and localization.
  - Red contours denote ground truth and cyan contours are false positive.
Self-driving car

• Recognizing traffic signs and objects;
Adversarial Cases

- Adversarial Cases are not simple ANN errors, but particular cases where the correct label can be easily classified by humans.
Self-Driving Uber Car Kills Pedestrian in Arizona, Where Robots Roam

A woman crossing Mill Avenue at its intersection with Curry Road in Tempe, Ariz., on Monday. A pedestrian was struck and killed by a self-driving Uber vehicle at the intersection a night earlier.

Caitlin O’Hara for The New York Times
Objectives

• Showing how unsafe an ANN through:
  • Generated adversarial cases
  • How adversarial is a set of images for the ANN neurons w.r.t. covering methods.
Artificial Neural Networks

• Artificial Neural Networks (ANNs) are versatile systems capable of generalizing and responding to unexpected inputs/patterns;
• ANNs are based in mathematical operations and learning algorithms.
Bounded Model Checking (BMC)

Basic idea: check negation of given property up to given depth

\[ \neg \varphi_0 \lor \neg \varphi_1 \lor \neg \varphi_2 \lor \ldots \lor \neg \varphi_{k-1} \lor \neg \varphi_k \]

transition system

counterexample trace

property

bound
Bounded Model Checking (BMC)

Basic idea: check negation of given property up to given depth

- Transition system $M$ unrolled $k$ times
  - for programs: loops, recursion, …
- Translated into verification condition $\psi$ such that

$$\psi \text{ satisfiable iff } \varphi \text{ has counterexample of max. depth } k$$
Bounded Model Checking (BMC)

Basic idea: check negation of given property up to given depth

• Transition system $M$ unrolled $k$ times
  – for programs: loops, recursion, …
• Translated into verification condition $\psi$ such that
  \[ \psi \text{ satisfiable iff } \varphi \text{ has counterexample of max. depth } k \]

BMC has been applied successfully to verify HW and SW
CUDA Operational Model

CUDA Documentation → Extract/Identify structure/properties → CUDA Operational Model

CUDA Operational Model → Adding assertions

CUDA Operational Model → Front-end

Front-end → ESBMC

ESBMC → Property violation

Property holds up to bound k → Verification Successful

CUDA Operational Model → Front-end

Front-end → ANN Source Code

ANN Source Code → Front-end

Front-end → Verification Successful
void feedForward()
{
    cublasSgemm(cublasHandle, CUBLAS_OP_T, CUBLAS_OP_N, 
    Layer.Outputs, batchSize, Layer.Inputs, 
    1, 
    Layer.MatrixBias, Layer.Inputs, 
    data, Layer.Inputs, 
    0, 
    LayerResults, Layer.Outputs);

    cublasSgemm(cublasHandle, CUBLAS_OP_N, CUBLAS_OP_N, 
    Layer.Outputs, batchSize, 1, 
    0, 
    BiasVector, Layer.Outputs, 
    onevec, 1, 
    1, 
    LayerResults, Layer.Outputs);

    __ESBMC_assert((LayerResults[CorrecLabel] > P) || 
        (LayerResults[WrongLabel] < P),"The correct label 
        became a wrong label")
Verification of Covering Methods

• For a ANN with linear activation function the neuron activation potencial \( v_{n,k} \), where \( n \) is the neuron index and \( k \) is the layer, can be obtained by:

\[
X_0 = -1
\]

\[
X_1
\]

\[
X_2
\]

\[
\begin{array}{l}
X_0 = -1 \\
X_1 \\
X_2
\end{array}
\]

\[
\begin{array}{c}
N_{1,1} \\
N_{1,2} \\
N_{1,3} \\
N_{2,1} \\
N_{2,2} \\
N_{3,1}
\end{array}
\]

\[
\begin{array}{c}
0.2 \\
0.4 \\
0.6 \\
0.5 \\
0.3 \\
0.6 \\
0.6 \\
0.7 \\
0.3 \\
0.4 \\
0.7 \\
0.2 \\
0.5 \\
0.8 \\
0.3 \\
0.8 \\
0.8 \\
0.3 \\
0.4 \\
0.7 \\
0.2 \\
0.5 \\
0.8 \\
-1 \\
-0.7 \\
0.1 \\
0.8 \\
0.1 \\
0.8 \\
0.1 \\
0.8 \\
0.1 \\
0.8 \\
0.1 \\
0.8 \\
0.1 \\
0.8 \\
0.1 \\
0.8 \\
0.1 \\
0.8 \\
0.1 \\
0.8 \\
0.1
\end{array}
\]

\[
\begin{array}{c|c|c}
\text{Input} & \text{Weight} & v \\
\hline
X_0 & W_{n0}^1 & X_0 W_{n0}^L + X_1 W_{n1}^L + X_2 W_{n2}^L \\
X_1 & W_{n1}^1 & \\
X_2 & W_{n2}^1 & \\
\end{array}
\]
Verification of Covering Methods

- Sign change (sc) occurs when the activation potential of a neuron changes with respect to two different inputs.
Verification of Covering Methods

• Sign change (sc):
  \[ X_A = \{X_1 = 1; X_2 = -3\} \]

\[
\begin{align*}
X_0 &= -1 \\
X_1 &\rightarrow N_{1,1} \\
X_2 &\rightarrow N_{1,1}
\end{align*}
\]

\[
\begin{align*}
X_1 &= 1 \\
X_2 &= -3
\end{align*}
\]

\[
\begin{array}{c|c|c}
\text{Input} & \text{Weight} & v_{1,3} \\
-1 & 0,4 & -1 \times 0,4 + 1 \times 0,8 -3 \times 0,3 = -0,5 \\
1 & 0,8 & \ \\
-3 & 0,3 & \\
\end{array}
\]
Verification of Covering Methods

- Sign change (sc):
  - $X_B = \{X_1 = 1; X_2 = -1\}$

### Table: Input and Weight Calculation

<table>
<thead>
<tr>
<th>Input</th>
<th>Weight</th>
<th>$v_{1,3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>0.4</td>
<td>-1*$0.4 + 1<em>0.8 -1</em>0.3 = 0.1$</td>
</tr>
<tr>
<td>1</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>0.3</td>
<td></td>
</tr>
</tbody>
</table>

*Diagram:*

- $X_0 = -1$
- $X_1$
- $X_2$
- $v_{1,3}$ calculation: $-1*0.4 + 1*0.8 -1*0.3 = 0.1$
Verification of Covering Methods

- **Sign change (sc):**
  - \( sc(n_1,3, X_A, X_B) = \text{true} \);

```plaintext
X_0 = -1

X_1

X_2
```

![Diagram with nodes and edges representing the covering methods with values 0.2, 0.4, 0.5, 0.6, 0.7, 0.8, -0.7, -0.3, 0.5, and 0.1.](attachment:diagram.png)
Verification of Covering Methods

- Value change (vc) occurs when there is no sign change in every neuron of a layer, but there is at least one who has a value change with respect to a metric h.
Verification of Covering Methods

• Value change (vc):
  - $X_A = \{X_1 = 1; X_2 = -1\}$

<table>
<thead>
<tr>
<th>Input</th>
<th>Weight</th>
<th>$v_{1,1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>0,2</td>
<td>-0,3</td>
</tr>
<tr>
<td>1</td>
<td>0,4</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>0,5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input</th>
<th>Weight</th>
<th>$v_{1,2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>0,3</td>
<td>-0,4</td>
</tr>
<tr>
<td>1</td>
<td>0,6</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>0,7</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input</th>
<th>Weight</th>
<th>$v_{1,3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>0,4</td>
<td>0,1</td>
</tr>
<tr>
<td>1</td>
<td>0,8</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>0,3</td>
<td></td>
</tr>
</tbody>
</table>
Verification of Covering Methods

• Value change (vc):
  \[ X_B = \{X_1 = 1; X_2 = -1.2\} \]

\[ X_0 = -1 \]

\[ X_1 \]

\[ X_2 \]

\[ \{-0.4, -0.054, 0.04\} \]

<table>
<thead>
<tr>
<th>Input</th>
<th>Weight</th>
<th>( v_{1,1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>0.2</td>
<td>-0.4</td>
</tr>
<tr>
<td>1</td>
<td>0.4</td>
<td>-0.54</td>
</tr>
<tr>
<td>-1.2</td>
<td>0.5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input</th>
<th>Weight</th>
<th>( v_{1,2} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>0.3</td>
<td>-0.54</td>
</tr>
<tr>
<td>1</td>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>-1.2</td>
<td>0.7</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input</th>
<th>Weight</th>
<th>( v_{1,3} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>0.4</td>
<td>0.04</td>
</tr>
<tr>
<td>1</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>-1.2</td>
<td>0.3</td>
<td></td>
</tr>
</tbody>
</table>
Verification of Covering Methods

- Value change (vc): Supposing the boolean \( h(x, y) = x - y > 0.13 \). Then, \( vc(h, n_{1,2}, X_A, X_B) = true \);
Verification of Covering Methods

- Distance change (dc) occurs when there is no sign change in every neuron of a layer, but all neurons shows a value change with respect to a metric g.
Verification of Covering Methods

• Distance change (dc):
  - $X_A = \{X_1 = 1; X_2 = -3\}$

\[ X_0 = -1 \]

\[ X_1 \]

\[ X_2 \]

\{-1.3, -1.8, -0.5\}

\[
\begin{array}{c|c|c}
\text{Input} & \text{Weight} & v_{1,1} \\
-1 & 0.2 & -1.3 \\
1 & 0.4 & \\
-3 & 0.5 & \\
\end{array}
\]

\[
\begin{array}{c|c|c}
\text{Input} & \text{Weight} & v_{1,2} \\
-1 & 0.3 & -1.8 \\
1 & 0.6 & \\
-3 & 0.7 & \\
\end{array}
\]

\[
\begin{array}{c|c|c}
\text{Input} & \text{Weight} & v_{1,3} \\
-1 & 0.4 & -0.5 \\
1 & 0.8 & \\
-3 & 0.3 & \\
\end{array}
\]
Verification of Covering Methods

• Distance change (dc):
  \( X_B = \{X_1 = 1; X_2 = -7\} \)

\[
\begin{array}{ccc}
\text{Input} & \text{Weight} & v_{1,1} \\
-1 & 0.2 & \text{-3.3} \\
1 & 0.4 & \\
-7 & 0.5 & \\
\end{array}
\]

\[
\begin{array}{ccc}
\text{Input} & \text{Weight} & v_{1,2} \\
-1 & 0.3 & \text{-4.6} \\
1 & 0.6 & \\
-7 & 0.7 & \\
\end{array}
\]

\[
\begin{array}{ccc}
\text{Input} & \text{Weight} & v_{1,3} \\
-1 & 0.4 & \text{-1.7} \\
1 & 0.8 & \\
-7 & 0.3 & \\
\end{array}
\]
Verification of Covering Methods

- Distance change (dc): Supposing the boolean \( g(l_k) = \text{EuclidianDistance}(l_k) > 0.1 \). Then, \( dc(g, l_1, X_A, X_B) = \text{true} \);

<table>
<thead>
<tr>
<th>( V'(X_1) )</th>
<th>( V'(X_2) )</th>
<th>( g(l_1) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.3</td>
<td>-3.3</td>
<td>True</td>
</tr>
<tr>
<td>-1.8</td>
<td>-4.6</td>
<td></td>
</tr>
<tr>
<td>-0.5</td>
<td>-1.7</td>
<td></td>
</tr>
</tbody>
</table>
Verification of Covering Methods

- Covering Methods are used to measure how adversarial a pair or a set of inputs to the ANN neurons. They are sectioned in four methods:

- **SS-Cover:**
  - \( sc(n_{k,i}, x_1, x_2) \);
  - \( \neg sc(n_{k,i}, x_1, x_2) \forall n_{k,l} \in L_k \);
  - \( sc(n_{k+1,j}, x_1, x_2) \);

- **DS-Cover:**
  - \( dc(g, k, x_1, x_2) \);
  - \( sc(n_{k+1,j}, x_1, x_2) \);

- **SV-Cover:**
  - \( sc(n_{k,i}, x_1, x_2) \);
  - \( \neg sc(n_{k,i}, x_1, x_2) \forall n_{k,l} \in L_k \);
  - \( vc(h, n_{k+1,j}, x_1, x_2) \);

- **DV-Cover:**
  - \( dc(g, k, x_1, x_2) \);
  - \( vc(h, n_{k+1,j}, x_1, x_2) \);
Verification of Covering Methods

• SS-Cover:
  \[ sc(n_{k,i}, x_1, x_2); \]
  \[ \neg sc(n_{k,i}, x_1, x_2) \forall n_{k,i} \in L_k; \]
  \[ sc(n_{k+1,j}, x_1, x_2); \]

- \( X_A = \{1, -1\} \)
- \( X_B = \{1, -3\} \)

<table>
<thead>
<tr>
<th></th>
<th>( x_A )</th>
<th>( x_B )</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_{1,1} )</td>
<td>-1,3</td>
<td>-0,3</td>
<td>F</td>
</tr>
<tr>
<td>( N_{1,2} )</td>
<td>-1,8</td>
<td>-0,4</td>
<td>F</td>
</tr>
<tr>
<td>( N_{1,3} )</td>
<td>-0,5</td>
<td>0,1</td>
<td>T</td>
</tr>
<tr>
<td>( N_{2,1} )</td>
<td>-0,79</td>
<td>0,51</td>
<td>T</td>
</tr>
<tr>
<td>( N_{2,2} )</td>
<td>-1,37</td>
<td>0,09</td>
<td>T</td>
</tr>
<tr>
<td>( N_{3,1} )</td>
<td>-1,41</td>
<td>0,35</td>
<td>T</td>
</tr>
</tbody>
</table>
Verification of Covering Methods

- SS-Cover:
  \[ sc(n_{k,i}, x_1, x_2); \]
  \[ \neg sc(n_{k,l}, x_1, x_2) \forall n_{k,l} \in L_k; \]
  \[ sc(n_{k+1,j}, x_1, x_2); \]

<table>
<thead>
<tr>
<th>Layer 1</th>
<th>X_A</th>
<th>X_B</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>N_{1,1}</td>
<td>-1,3</td>
<td>-0,3</td>
<td>F</td>
</tr>
<tr>
<td>N_{1,2}</td>
<td>-1,8</td>
<td>-0,4</td>
<td>F</td>
</tr>
<tr>
<td>N_{1,3}</td>
<td>-0,5</td>
<td>0,1</td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Layer 2</th>
<th>X_A</th>
<th>X_B</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>N_{2,1}</td>
<td>-0,79</td>
<td>0,51</td>
<td>T</td>
</tr>
<tr>
<td>N_{2,2}</td>
<td>-1,37</td>
<td>0,09</td>
<td>T</td>
</tr>
</tbody>
</table>
Verification of Covering Methods

• SS-Cover:
  \[ sc(n_{k,i}, x_1, x_2); \]
  \[ \neg sc(n_{k,l}, x_1, x_2) \forall n_{k,l} \in L_k; \]
  \[ sc(n_{k+1,j}, x_1, x_2); \]

<table>
<thead>
<tr>
<th>Layer1</th>
<th>( X_A )</th>
<th>( X_B )</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_{1,1} )</td>
<td>-1.3</td>
<td>-0.3</td>
<td>F</td>
</tr>
<tr>
<td>( N_{1,2} )</td>
<td>-1.8</td>
<td>-0.4</td>
<td>F</td>
</tr>
<tr>
<td>( N_{1,3} )</td>
<td>-0.5</td>
<td>0.1</td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Layer2</th>
<th>( X_A )</th>
<th>( X_B )</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_{2,1} )</td>
<td>-0.79</td>
<td>0.51</td>
<td>T</td>
</tr>
<tr>
<td>( N_{2,2} )</td>
<td>-1.37</td>
<td>0.09</td>
<td>T</td>
</tr>
</tbody>
</table>
Verification of Covering Methods

- **SS-Cover:**
  \[ sc(n_{k,i}, x_1, x_2); \]
  \[ \neg sc(n_{k,l}, x_1, x_2) \forall n_{k,l} \in L_k; \]
  \[ sc(n_{k+1,j}, x_1, x_2); \]

<table>
<thead>
<tr>
<th>Layer1</th>
<th>X_A</th>
<th>X_B</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>N_{1,1}</td>
<td>-1,3</td>
<td>-0,3</td>
<td>F</td>
</tr>
<tr>
<td>N_{1,2}</td>
<td>-1,8</td>
<td>-0,4</td>
<td>F</td>
</tr>
<tr>
<td>N_{1,3}</td>
<td>-0,5</td>
<td>0,1</td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Layer2</th>
<th>X_A</th>
<th>X_B</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>N_{2,1}</td>
<td>-0,79</td>
<td>0,51</td>
<td>T</td>
</tr>
<tr>
<td>N_{2,2}</td>
<td>-1,37</td>
<td>0,09</td>
<td>T</td>
</tr>
</tbody>
</table>
Verification of Covering Methods

- SS-Cover:
  \[ \text{sc}(n_{k,i}, x_1, x_2) \]
  \[ \neg \text{sc}(n_{k,l}, x_1, x_2) \forall n_{k,l} \in L_k \]
  \[ \text{sc}(n_{k+1,j}, x_1, x_2) \]

The neuron pair \( \{N_{1,3}, N_{2,1}\} \) is SS-covered by \( X_A \) and \( X_B \)

<table>
<thead>
<tr>
<th>Layer1</th>
<th>( X_A )</th>
<th>( X_B )</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_{1,1} )</td>
<td>-1,3</td>
<td>-0,3</td>
<td>F</td>
</tr>
<tr>
<td>( N_{1,2} )</td>
<td>-1,8</td>
<td>-0,4</td>
<td>F</td>
</tr>
<tr>
<td>( N_{1,3} )</td>
<td>-0,5</td>
<td>0,1</td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Layer2</th>
<th>( X_A )</th>
<th>( X_B )</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_{2,1} )</td>
<td>-0.79</td>
<td>0.51</td>
<td>T</td>
</tr>
<tr>
<td>( N_{2,2} )</td>
<td>-1.37</td>
<td>0.09</td>
<td>T</td>
</tr>
</tbody>
</table>
Verification of Covering Methods

• DV-Cover:
  \( dc(g, k, x_1, x_2); \)
  \( vc(h, n_{k+1,j}, x_1, x_2); \)

\( X_A = \{1, -3\} \)
\( X_B = \{1, -7\} \)

<table>
<thead>
<tr>
<th></th>
<th>( X_A )</th>
<th>( X_B )</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_{1,1} )</td>
<td>-1,3</td>
<td>-3,3</td>
<td>F</td>
</tr>
<tr>
<td>( N_{1,2} )</td>
<td>-1,8</td>
<td>-4,6</td>
<td>F</td>
</tr>
<tr>
<td>( N_{1,3} )</td>
<td>-0,5</td>
<td>-1,7</td>
<td>F</td>
</tr>
<tr>
<td>( N_{2,1} )</td>
<td>-0,79</td>
<td>-3,39</td>
<td>F</td>
</tr>
<tr>
<td>( N_{2,2} )</td>
<td>-1,37</td>
<td>-4,29</td>
<td>F</td>
</tr>
<tr>
<td>( N_{3,1} )</td>
<td>-1,41</td>
<td>-4,95</td>
<td>F</td>
</tr>
</tbody>
</table>
**Verification of Covering Methods**

- DV-Cover:
  \[ dc(g, k, x_1, x_2); \]
  \[ vc(h, n_{k+1,j}, x_1, x_2); \]

<table>
<thead>
<tr>
<th>Layer1</th>
<th>( X_A )</th>
<th>( X_B )</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_{1,1} )</td>
<td>-1,3</td>
<td>-3,3</td>
<td>F</td>
</tr>
<tr>
<td>( N_{1,2} )</td>
<td>-1,8</td>
<td>-4,6</td>
<td>F</td>
</tr>
<tr>
<td>( N_{1,3} )</td>
<td>-0,5</td>
<td>-1,7</td>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Layer2</th>
<th>( X_A )</th>
<th>( X_B )</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_{2,1} )</td>
<td>-0,79</td>
<td>-3,39</td>
<td>F</td>
</tr>
<tr>
<td>( N_{2,2} )</td>
<td>-1,37</td>
<td>-4,29</td>
<td>F</td>
</tr>
</tbody>
</table>
Verification of Covering Methods

- DV-Cover:
  \[ dc(g, k, x_1, x_2); \]
  \[ vc(h, n_{k+1,j}, x_1, x_2); \]

Layer 1

<table>
<thead>
<tr>
<th>Layer 1</th>
<th>( X_A )</th>
<th>( X_B )</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_{1,1} )</td>
<td>-1.3</td>
<td>-3.3</td>
<td>F</td>
</tr>
<tr>
<td>( N_{1,2} )</td>
<td>-1.8</td>
<td>-4.6</td>
<td>F</td>
</tr>
<tr>
<td>( N_{1,3} )</td>
<td>-0.5</td>
<td>-1.7</td>
<td>F</td>
</tr>
</tbody>
</table>

Layer 2

<table>
<thead>
<tr>
<th>Layer 2</th>
<th>( X_A )</th>
<th>( X_B )</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_{2,1} )</td>
<td>-0.79</td>
<td>-3.39</td>
<td>F</td>
</tr>
<tr>
<td>( N_{2,2} )</td>
<td>-1.37</td>
<td>-4.29</td>
<td>F</td>
</tr>
</tbody>
</table>
Verification of Covering Methods

- DV-Cover:
  \[ \text{dc}(g, k, x_1, x_2); \]
  \[ \text{vc}(h, n_{k+1,j}, x_1, x_2); \]

The pair \{N_{1,3}, N_{2,2}\} is DV-covered by \(X_A\) and \(X_B\)

**Layer 1**

<table>
<thead>
<tr>
<th></th>
<th>(X_A)</th>
<th>(X_B)</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>(N_{1,1})</td>
<td>-1,3</td>
<td>-3,3</td>
<td>F</td>
</tr>
<tr>
<td>(N_{1,2})</td>
<td>-1,8</td>
<td>-4,6</td>
<td>F</td>
</tr>
<tr>
<td>(N_{1,3})</td>
<td>-0,5</td>
<td>-1,7</td>
<td>F</td>
</tr>
</tbody>
</table>

**Layer 2**

<table>
<thead>
<tr>
<th></th>
<th>(X_A)</th>
<th>(X_B)</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>(N_{2,1})</td>
<td>-0.79</td>
<td>-3.39</td>
<td>F</td>
</tr>
<tr>
<td>(N_{2,2})</td>
<td>-1,37</td>
<td>-4.29</td>
<td>F</td>
</tr>
</tbody>
</table>
Verification of Adversarial Cases

• Conditions for adversarial cases:

\[ I_d \in D^{m \times n}, \ I \in M^{m \times n}, \ D^{m \times n} \subseteq M^{m \times n} \]

• Restriction:

\[ R = \{ I \in M^{mn} \mid \delta(I_d, I) \leq b \} \]

• Adversarial case definition:

\[ \mathcal{Y}^d = \{ N(I), \ \forall I \in R \} \]
Verification of Adversarial Cases

Assume the euclidian distance

Non-Deterministic

Base image

$R = \{ I \in M^{mn} | \delta(I^d, I) \leq b \}$

Checks if output corresponds to the base image label

$Y^d = \{ N(I) , \forall I \in R \}$

ANN model
Experimental Evaluation

Benchmark Description:
- ANN that solves the problem of vocalic recognition;
- Dataset composed by: 100 vocalics with noises; 100 characters;
- Backpropagation
- Cross-validation
Experimental Evaluation

- Experiments were conducted on a 8-core 3.40GHz Intel Core i7 with 24 GB of RAM and Linux OS.
- Frameworks versions:
  - CUDA v9.0, cuDNN v5.0, cuBLAS v10.1, and ESBMC-GPU v2.0.
Experimental Objectives

1. Evaluate the performance and correctness of our symbolic verification algorithms to check all four covering methods;
2. Evaluate the performance and correctness of our verification algorithm checkNN to verify adversarial cases obtained from changing input images and parameter proximity.
Results

- The four properties specify that 80% of all neurons must be covered by a set of inputs.
- Verification time of the dataset w.r.t all 4 covering methods is around 20 minutes. It correctly verified the 4 properties.
- The verification of two inputs did not take more than a few seconds.
Results

- Adversarial cases for label “E” missclassified as label “O”:

  ![Image](image1)
  
  - $b = 0.5$
  - $b = 1.5$
  - $b = 2.5$
  - $b = 3.5$

- Label “U” missclassified as label “O”:

  ![Image](image2)
  
  - $b = 0.3$
  - $b = 0.5$
  - $b = 1.0$
  - $b = 1.5$
## Results

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Image</th>
<th>$\lambda$</th>
<th>Verification Time (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ex1</td>
<td>Vocalic O</td>
<td>0.5</td>
<td>1</td>
</tr>
<tr>
<td>Ex2</td>
<td>Vocalic O</td>
<td>1.5</td>
<td>4</td>
</tr>
<tr>
<td>Ex3</td>
<td>Vocalic O</td>
<td>2.5</td>
<td>8</td>
</tr>
<tr>
<td>Ex4</td>
<td>Vocalic O</td>
<td>3.5</td>
<td>6</td>
</tr>
<tr>
<td>Ex5</td>
<td>Vocalic E</td>
<td>0.5</td>
<td>25</td>
</tr>
<tr>
<td>Ex6</td>
<td>Vocalic E</td>
<td>0.7</td>
<td>25</td>
</tr>
<tr>
<td>Ex7</td>
<td>Vocalic E</td>
<td>1.5</td>
<td>14</td>
</tr>
<tr>
<td>Ex8</td>
<td>Vocalic E</td>
<td>3.0</td>
<td>12</td>
</tr>
<tr>
<td>Ex9</td>
<td>Vocalic U</td>
<td>0.3</td>
<td>6</td>
</tr>
<tr>
<td>Ex10</td>
<td>Vocalic U</td>
<td>0.5</td>
<td>5</td>
</tr>
<tr>
<td>Ex11</td>
<td>Vocalic U</td>
<td>1.0</td>
<td>20</td>
</tr>
<tr>
<td>Ex12</td>
<td>Vocalic U</td>
<td>1.5</td>
<td>19</td>
</tr>
<tr>
<td>Ex13</td>
<td>Vocalic A</td>
<td>1.0</td>
<td>63</td>
</tr>
<tr>
<td>Ex14</td>
<td>Vocalic A</td>
<td>1.5</td>
<td>63</td>
</tr>
</tbody>
</table>
Conclusion

• Our verification method is able to find adversarial cases for different input images and proximity parameter values;
• Our approach exhaustively verifies all possible adversarial cases inside the proximity parameter b;
• The verification of covering methods is able to verify our dataset correctly. The dataset is not able to cover 80% of neuron w.r.t. a covering method;
• There is a high verification time in some benchmarks due to bitaccurate verification.
Future Work

• Support convolutional layers;
• Implement some techniques as invariant inference to prune the state space exploration;
• Investigate fault localization and repair techniques;
Questions?

Luiz Henrique Coelho Sena
lhcs@icomp.ufam.edu.br
Universidade Federal do Amazonas