Formal Methods for Robust Artificial Intelligence
State of the Art

Caterina Urban
ANTIQUE Research Team, Inria & École Normale Supérieure | Université PSL
Artificial Intelligence Development Process

Artificial Intelligence Pipeline

1. data
2. data preparation
3. model training
4. model deployment
5. predictions
Model Training is **Highly Non-Deterministic**

*Image with cartoon character digging a pile of data and questions about machine learning.*

- **Data Preparation**
- **Model Training**
- **Model Deployment**
- **Predictions**

*Note: No predictability and traceability.*
Models Only Give Probabilistic Guarantees

Data preparation model training model deployment predictions

not sufficient for guaranteeing an acceptable failure rate under any circumstances

Stop + Max Speed 100

Models Only Give Probabilistic Guarantees

not sufficient for guaranteeing an acceptable failure rate under any circumstances

Models Only Give Probabilistic Guarantees

not sufficient for guaranteeing an acceptable failure rate under any circumstances
A self-driving Uber ran a red light last December, contrary to company claims

Feds Say Self-Driving Uber SUV Did Not Recognize Jaywalking Pedestrian In Fatal Crash
Formal Methods

Mathematical Guarantees of Safety

Deductive Verification
- extremely expressive
- relies on the user to guide the proof

Model Checking
- analysis of a model of the software
- sound and complete with respect to the model

Static Analysis
- analysis of the software at some level of abstraction
- fully automatic and sound by construction
- generally not complete
Neural Network Models
Feed-Forward Neural Networks

Fully-Connected Layers with ReLU Activation Functions

\[ x_{i,j} = \max \left\{ 0, \sum_k w_{j,k}^{i-1} \cdot x_{i-1,k} + b_{i,j} \right\} \]
Safety Verification

\[ l_j \leq x_{0,j} \leq u_j \]

\[ x_{i,j} = \max \left\{ 0, \sum_{k} w^{i-1}_{j,k} \cdot x_{i-1,k} + b_{i,j} \right\} \]

\[ x_N > 0 \]
Model Checking Methods
SMT-Based Methods

Safety Verification Reduced to Constraint Satisfiability

\[ l_j \leq x_{0,j} \leq u_j \]
\[ j \in \{0, \ldots, |X_0|\} \]

\[ \hat{x}_{i+1,j} = \sum_{k=0}^{|X_i|} w_{j,k}^i \cdot x_{i,k} + b_{i,j} \]
\[ i \in \{0, \ldots, n-1\} \]

\[ x_{i,j} = \max\{0, \hat{x}_{i,j}\} \]
\[ i \in \{1, \ldots, n-1\}, j \in \{0, \ldots, |X_i|\} \]

\[ x_N \leq 0 \]

input specification

output specification

satisfiable \(\rightarrow\) counterexample
otherwise \(\rightarrow\) safe
Planet

\[ x_{i,j} = \max\{0, \hat{x}_{i,j}\} \]

\[
0 \leq x_{i,j} \\
\hat{x}_{i,j} \leq x_{i,j} \\
x_{i,j} \leq \frac{b_{i,j}}{b_{i,j} - a_{i,j}} \cdot (\hat{x}_{i,j} - a_{i,j})
\]

R. Ehlers - Formal Verification of Piece-Wise Linear Feed-Forward Neural Networks (ATVA 2017)
Reluplex

Based on the simplex algorithm extended to support ReLU constraints

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{00}$</td>
<td>$v_{00}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$\tilde{x}_{ij}$</td>
<td>$\tilde{v}_{ij}$</td>
</tr>
<tr>
<td>$x_{ij}$</td>
<td>$v_{ij}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$x_{N}$</td>
<td>$v_{N}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{x}_{ij}$</td>
<td>$\hat{v}_{ij}$</td>
</tr>
<tr>
<td>$x_{ij}$</td>
<td>$v_{ij}$</td>
</tr>
<tr>
<td>$x_{N}$</td>
<td>$v_{N}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{00}$</td>
<td>$v_{00}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$\hat{x}_{ij}$</td>
<td>$\hat{v}_{ij}$</td>
</tr>
<tr>
<td>$x_{ij}$</td>
<td>$v_{ij}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$x_{N}$</td>
<td>$v_{N}$</td>
</tr>
</tbody>
</table>

G. Katz et al. - Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks (CAV 2017)
**Reluplex**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{00}$</td>
<td>$v_{00}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$\hat{x}_{ij}$</td>
<td>$\hat{v}_{ij}$</td>
</tr>
<tr>
<td>$x_{ij}$</td>
<td>$v_{ij}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$x_N$</td>
<td>$v_N$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{x}_{ij}$</td>
<td>$\hat{v}_{ij}$</td>
</tr>
<tr>
<td>$x_{ij}$</td>
<td>$v_{ij}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$x_N$</td>
<td>$v_N$</td>
</tr>
</tbody>
</table>

**Follow-up Work**

G. Katz et al. - The Marabou Framework for Verification and Analysis of Deep Neural Networks (CAV 2019)

---

G. Katz et al. - Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks (CAV 2017)
Other SMT-Based Methods

- L. Pulina and A. Tacchella - An Abstraction-Refinement Approach to Verification of Artificial Neural Networks (CAV 2010)
  the first formal verification method for neural networks

  an approach for finding the nearest adversarial example according to the $L_\infty$ distance

- X. Huang, M. Kwiatkowska, S. Wang, and M. Wu - Safety Verification of Deep Neural Networks (CAV 2017)
  an approach for proving local robustness to adversarial perturbations

  approaches focusing on binarized neural networks
MILP-Based Methods

Safety Verification Reduced to Mixed Integer Linear Program

\[ l_j \leq x_{0,j} \leq u_j \]
\[ j \in \{0,\ldots,|X_0|\} \]

\[ h_{i+1,j} = \sum_{k=0}^{\frac{|X_i|}{|X_i|}} w_{j,k}^i \cdot x_{i,k} + b_{i,j} \]
\[ i \in \{0,\ldots,n-1\} \]

\[ x_{i,j} = \delta_{i,j} \cdot \hat{x}_{i,j} \]
\[ \delta_{i,j} \in \{0,1\} \]

\[ \delta_{i,j} = 1 \Rightarrow \hat{x}_{i,j} \geq 0 \]
\[ i \in \{1,\ldots,n-1\} \]
\[ j \in \{0,\ldots,|X_i|\} \]

\[ \delta_{i,j} = 0 \Rightarrow \hat{x}_{i,j} < 0 \]

\[ \min x_N \]

\[ \min x_N \leq 0 \rightarrow \text{counterexample} \]

\[ \text{otherwise} \rightarrow \text{safe} \]
MILP-Based Methods

Bounded MILP Encoding with Symmetric Bounds

\[
\hat{x}_{i+1,j} = \sum_{k=0}^{\lvert X_i \rvert} w_{j,k}^i \cdot x_{i,k} + b_{i,j} \\
0 \leq x_{i,j} \leq M_{i,j} \cdot \delta_{i,j} \\
\hat{x}_{i,j} \leq x_{i,j} \leq \hat{x}_{i,j} - M_{i,j} \cdot (1 - \delta_{i,j}) \\
M_{i,j} = \max\{-l_i, u_i\}
\]

\( i \in \{0, \ldots, n - 1\} \)

\( \delta_{i,j} \in \{0,1\} \)

\( i \in \{1, \ldots, n - 1\} \)

\( j \in \{0, \ldots, \lvert X_i \rvert\} \)
Output Range Analysis

\[ l_j \leq x_{0,j} \leq u_j \]

\[ \hat{x}_{i+1,j} = \sum_{k=0}^{X_i} w_{j,k}^i \cdot x_{i,k} + b_{i,j} \]

\[ 0 \leq x_{i,j} \leq M_{i,j} \cdot \delta_{i,j} \]

\[ \hat{x}_{i,j} \leq x_{i,j} \leq \hat{x}_{i,j} - M_{i,j} \cdot (1 - \delta_{i,j}) \]

\[ M_{i,j} = \max\{-l_i, u_i\} \]

\[ \min \ x_N \]

S. Dutta et al. - Output Range Analysis for Deep Feedforward Neural Networks (NFM 2018)
**Output Range Analysis**

\[ l_j \leq x_{0,j} \leq u_j \]

\[ \hat{x}_{i+1,j} = \sum_{k=0}^{X_i} w_{j,k}^i \cdot x_{i,k} + b_{i,j} \]

\[ 0 \leq x_{i,j} \leq M_{i,j} \cdot \delta_{i,j} \]

\[ \hat{x}_{i,j} \leq x_{i,j} \leq \hat{x}_{i,j} - M_{i,j} \cdot (1 - \delta_{i,j}) \]

\[ M_{i,j} = \max \{-l_i, u_i\} \]

\[ \min x_N \]

---

S. Dutta et al. - Output Range Analysis for Deep Feedforward Neural Networks (NFM 2018)
Output Range Analysis

\[ l_j \leq x_{0,j} \leq u_j \]

\[ \hat{x}_{i+1,j} = \sum_{k=0}^{\vert X_i \vert} w^i_{j,k} \cdot x_{i,k} + b_{i,j} \]

\[ 0 \leq x_{i,j} \leq M_{i,j} \cdot \delta_{i,j} \]

\[ \hat{x}_{i,j} \leq x_{i,j} \leq \hat{x}_{i,j} - M_{i,j} \cdot (1 - \delta_{i,j}) \]

\[ M_{i,j} = \max\{-l_i, u_i\} \]

\[ x_N < L \]

S. Dutta et al. - Output Range Analysis for Deep Feedforward Neural Networks (NFM 2018)
Output Range Analysis

\[ l_j \leq x_{0,j} \leq u_j \]

\[ \hat{x}_{i+1,j} = \sum_{k=0}^{\lfloor X_i \rfloor} w_{i,k} \cdot x_{i,k} + b_{i,j} \]

\[ 0 \leq x_{i,j} \leq M_{i,j} \cdot \delta_{i,j} \]

\[ \hat{x}_{i,j} \leq x_{i,j} \leq \hat{x}_{i,j} - M_{i,j} \cdot (1 - \delta_{i,j}) \]

\[ M_{i,j} = \max\{-l_i, u_i\} \]

\[ x_N < L \]

S. Dutta et al. - Output Range Analysis for Deep Feedforward Neural Networks (NFM 2018)
Output Range Analysis

\[ l_j \leq x_{0,j} \leq u_j \]

\[ \hat{x}_{i+1,j} = \sum_{k=0}^{\vert X_i \vert} w_{j,k}^i \cdot x_{i,k} + b_{i,j} \]

\[ 0 \leq x_{i,j} \leq M_{i,j} \cdot \delta_{i,j} \]

\[ \hat{x}_{i,j} \leq x_{i,j} \leq \hat{x}_{i,j} - M_{i,j} \cdot (1 - \delta_{i,j}) \]

\[ M_{i,j} = \max\{-l_i, u_i\} \]

\[ x_N < \hat{L} \]

S. Dutta et al. - Output Range Analysis for Deep Feedforward Neural Networks (NFM 2018)
Output Range Analysis

\[ l_j \leq x_{0,j} \leq u_j \]

\[ \hat{x}_{i+1,j} = \sum_{k=0}^{\left| X_i \right|} w_{j,k}^i \cdot x_{i,k} + b_{i,j} \]

\[ 0 \leq x_{i,j} \leq M_{i,j} \cdot \delta_{i,j} \]

\[ \hat{x}_{i,j} \leq x_{i,j} \leq \hat{x}_{i,j} - M_{i,j} \cdot (1 - \delta_{i,j}) \]

\[ M_{i,j} = \max\{-l_i, u_i\} \]

\[ x_N < \hat{L} \]

S. Dutta et al. - Output Range Analysis for Deep Feedforward Neural Networks (NFM 2018)
MILP-Based Methods
Bounded MILP Encoding with Asymmetric Bounds

\[
\hat{x}_{i+1,j} = \sum_{k=0}^{\left|X_i\right|} w_{j,k}^i \cdot x_{i,k} + b_{i,j} \\
0 \leq x_{i,j} \leq u_{i,j} \cdot \delta_{i,j} \\
\hat{x}_{i,j} \leq x_{i,j} \leq \hat{x}_{i,j} - l_{i,j} \cdot (1 - \delta_{i,j})
\]

\( i \in \{0,\ldots,n-1\} \)

\( \delta_{i,j} \in \{0,1\} \)

\( i \in \{1,\ldots,n-1\} \)

\( j \in \{0,\ldots,\left|X_i\right|\} \)
MIPVerify
Finding Nearest Adversarial Example

\[
\min_{X'} \ d(X, X')
\]

\[
\hat{x}_{i+1,j} = \sum_{k=0}^{X_i} w_{j,k}^i \cdot x_{i,k} + b_{i,j}
\]

\[
0 \leq x_{i,j} \leq u_{i,j} \cdot \delta_{i,j}
\]

\[
\hat{x}_{i,j} \leq x_{i,j} \leq \hat{x}_{i,j} - l_{i,j} \cdot (1 - \delta_{i,j})
\]

\[x_N \neq 0\]

V. Tjeng, K. Xiao, and R. Tedrake - Evaluating Robustness of Neural Networks with Mixed Integer Programming (ICLR 2019)
Other MILP-Based Methods

• R. Bunel, I. Turkaslan, P. H. S. Torr, P. Kohli, and M. P. Kumar - A Unified View of Piecewise Linear Neural Network Verification (NeurIPS 2018)
  a unifying verification framework for piecewise-linear ReLU neural networks

  an approach for finding a lower bound on robustness to adversarial perturbations

• M. Fischetti and J. Jo - Deep Neural Networks and Mixed Integer Linear Optimization (2018)
  an approach for feature visualization and building adversarial examples
Static Analysis Methods
Abstract Interpretation-Based Methods

1. **proceed forwards from an abstraction of the input specification**

2. **check output for inclusion in output specification**: included → ✔️ safe otherwise → 🚨 alarm
Symbolic Propagation

\[ x_{i,j} \mapsto \begin{cases} \sum_{k=0}^{i-1} c_k \cdot x_k + c_k, c \in \mathcal{R} | x_k | \\ [a, b] \end{cases} \]

represent each neuron as a linear combination of the inputs and the ReLUs in previous layers

\[ x_{i-1,0} \mapsto E_{i-1,0} \]

\[ \cdots \]

\[ x_{i-1,j} \mapsto E_{i-1,j} \]

\[ \cdots \]

\[ x_{i,j} \mapsto \begin{cases} E_{i,j} \\ [a, b] \end{cases} \]

ReLU

\[ x_{i,j} \mapsto \begin{cases} E_{i,j} \\ [a, b] \end{cases} \]

\[ x_{i,j} \mapsto \begin{cases} x_{i,j} \\ [0, b] \end{cases} \]

\[ x_{i,j} \mapsto \begin{cases} 0 \\ [0, 0] \end{cases} \]

J. Li et al. - Analyzing Deep Neural Networks with Symbolic Propagation (SAS 2019)
DeepPoly

\[ x_{i+1,j} \mapsto \begin{cases} 
[ \sum_k c_{i,k} \cdot x_{i,k} + c, \sum_k d_{i,k} \cdot x_{i,k} + d ] & c_{i,k}, c, d_{i,k}, d \in \mathcal{R} \\
[a, b] & a, b \in \mathcal{R}
\end{cases} \]

\[ x_{i,j} \mapsto \begin{cases} 
[L_{i,j}, U_{i,j}] & [a, b] \\
[0, 0] & [0, 0]
\end{cases} \]

G. Singh, T. Gehr, M. Püschel, and M. Vechev - An Abstract Domain for Certifying Neural Networks (POPL 2019)
Other Abstract Interpretation Methods

- T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. Vechev - AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation (S&P 2018)
  the first use of abstract interpretation for verifying neural networks

- G. Singh, T. Gehr, M. Mirman, M. Püschel, and M. Vechev - Fast and Effective Robustness Certification (NeurIPS 2018)
  a custom zonotope domain for certifying neural networks

  a framework to jointly approximate $k$ ReLU activations

- C. Urban, M. Christakis, V. Wüstholz, and F. Zhang - Perfectly Parallel Fairness Certification of Neural Networks (OOPSLA 2020)
  an approach for verifying fairness of neural network classifiers for tabular data
Other Complete Methods
Star Sets

Exact Static Analysis Method

\[ \Theta \overset{\text{def}}{=} \langle c, V, P \rangle \]

\( c \in \mathbb{R}^n \): center

\( V = \{ v_1, \ldots, v_m \} \): basis vectors in \( \mathbb{R}^n \)

\( P: \mathbb{R}^m \rightarrow \{ \bot, \top \} \): predicate

\[ \llbracket \Theta \rrbracket = \{ x \mid x = c + \sum_{i=1}^{m} \alpha_i v_i \text{ such that } P(\alpha_1, \ldots, \alpha_m) = \top \} \]

- fast and cheap affine mapping operations \( \rightarrow \) neural network layers
- inexpensive intersections with half-spaces \( \rightarrow \) ReLU activations

H.-D. Tran et al. - Star-Based Reachability Analysis of Deep Neural Networks (FM 2018)
Star Sets

Exact Static Analysis Method

\[ \Theta \overset{\text{def}}{=} \langle c, V, P \rangle \]

- \( c \in \mathbb{R}^n \): center
- \( V = \{v_1, \ldots, v_m\} \): basis vectors in \( \mathbb{R}^n \)
- \( P: \mathbb{R}^m \rightarrow \{\bot, T\} \): predicate

\[ [\Theta] = \{ x | x = c + \sum_{i=1}^{m} \alpha_i v_i \text{ such that } P(\alpha_1, \ldots, \alpha_m) = T \} \]

- fast and cheap **affine mapping operations** → neural network layers
- inexpensive **intersections with half-spaces** → ReLU activations

---

H.-D. Tran et al. - Star-Based Reachability Analysis of Deep Neural Networks (FM 2018)

Follow-up Work

H.-D. Tran et al. - Verification of Deep Convolutional Neural Networks Using ImageStars (CAV 2020)
ReluVal
Asymptotically Complete Method

use symbolic propagation
+ iterative input refinement

S. Wang et al. - Formal Security Analysis of Neural Networks Using Symbolic Intervals (USENIX Security 2018)
Neurify
Asymptotically Complete Method

\[
x_{i,j} \mapsto \begin{cases} 
  [ \sum_k c_{0,k} \cdot x_{0,k} + c, \ \sum_k d_{0,k} \cdot x_{0,k} + d ] 
  & c_{0,k}, c, d_{0,k}, d \in \mathcal{R} \\
  [a, b] & a, b \in \mathcal{R}
\end{cases}
\]

\[
x_{i,j} \mapsto \begin{cases} 
  [E_{i,j}, E_{i,j}] 
  & 0 \leq a \\
  [a, b] & a < 0 \land 0 < b \\
  [0, 0] & b \leq 0
\end{cases}
\]

Other Complete Methods

• W. Ruan, X. Huang, and Marta Kwiatkowska - Reachability Analysis of Deep Neural Networks with Provable Guarantees (IJCAI 2018)
  a global optimization-based approach for verifying Lipschitz continuous neural networks

• G. Singh, T. Gehr, M. Püschel, and M. Vechev - Boosting Robustness Certification of Neural Networks (ICLR 2019)
  an approach combining abstract interpretation and (mixed integer) linear programming
Other Incomplete Methods
Interval Neural Networks

Abstraction-Based Method

merge neurons layer-wise according to partitioning strategy and replace weights with intervals

\[ l_j \leq x_{0,j} \leq u_j \]

P. Prabhakar and Z. R. Afza - Abstraction based Output Range Analysis for Neural Networks (NeurIPS 2019)
Interval Neural Networks

Abstraction-Based Method

\[ l_j \leq x_{0,j} \leq u_j \]

\[ x_N > 0 \]

Related Work

Y. Y. Elboher et al. - An Abstraction-Based Framework for Neural Network Verification (CAV 2020)

P. Prabhakar and Z. R. Afza - Abstraction based Output Range Analysis for Neural Networks (NeurIPS 2019)
Other Incomplete Methods

• W. Xiang, H.-D. Tran, and T. T. Johnson - Output Reachable Set Estimation and Verification for Multi-Layer Neural Networks (2018)
  an approach combining simulation and linear programming

• K. Dvijotham, R. Stanforth, S. Gowal, T. Mann, and P. Kohli - A Dual Approach to Scalable Verification of Deep Networks (UAI 2018)
  an approach based on duality for verifying neural networks

• E. Wong and Z. Kolter - Provable Defenses Against Adversarial Examples via the Convex Outer Adversarial Polytope (ICML 2018)
  A. Raghunathan, J. Steinhardt, and P. Liang - Certified Defenses against Adversarial Examples (ICML 2018)
  Towards Fast Computation of Certified Robustness for ReLU Networks (ICML 2018)
  Robustness Certification with General Activation Functions (NeurIPS 2018)
  approaches for finding a lower bound on robustness to adversarial perturbations
Other Incomplete Methods

  approach focusing on convolutional neural networks

  H. Zhang, M. Shinn, A. Gupta, A. Gurfinkel, N. Le, and N. Narodytska - Verification of Recurrent Neural Networks for Cognitive Tasks via Reachability Analysis (ECAI 2020)  
  approaches focusing on recurrent neural networks

• D. Gopinath, H. Converse, C. S. Pasareanu, and A. Taly - Property Inference for Deep Neural Networks (ASE 2019)  
  an approach for inferring safety properties of neural networks
### Complete Methods

**ADVANTAGES**
- sound and complete

**DISADVANTAGES**
- soundness not typically guaranteed with respect to floating-point arithmetic
- do not scale to large models
- often limited to certain model architectures

### Incomplete Methods

**ADVANTAGES**
- able to scale to large models
- sound often also with respect to floating-point arithmetic
- less limited to certain model architectures

**DISADVANTAGES**
- suffer from false positives