Incremental Verification of Neural Networks Guided by Counterexample Potentiality

SE4AI: Deep learning model formal verification

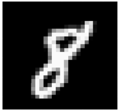
Guanqin Zhang

Supervisor: AP.Yulei Sui[1], Dr. Dilum Bandara[2], Dr.Shiping Chen[2]

[1]UNSW School of Computer Science and Engineering
[2]CSIRO DATA61

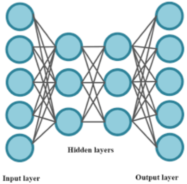
January 9, 2024

Adversarial Input Perturbation



x_0

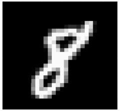
Nerual Network N



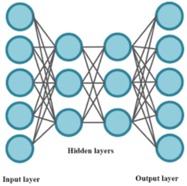
Label 8

Adversarial Input Perturbation

Nerual Network N

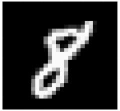


x_0

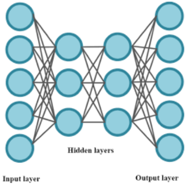


Label 8

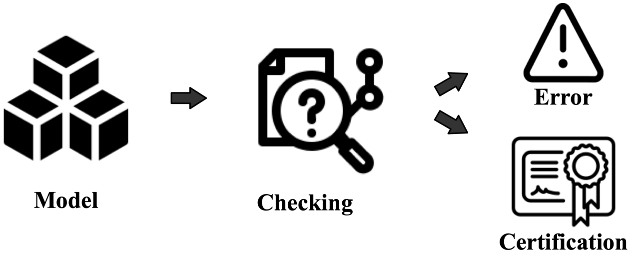
Nerual Network N



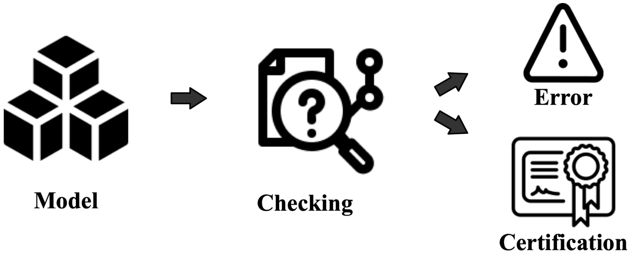
$x' \in (l_\infty, \epsilon)$



Label 6



(1) Given: A network model $N : \mathbb{R}^n \rightarrow \mathbb{R}^m$,
(l_p, ϵ)-adversary region, i.e., $\{\boldsymbol{x}' \mid \|\boldsymbol{x}' - \boldsymbol{x}_0\|_p \leq \epsilon\}, p = \infty$



(1) Given: A network model $N : \mathbb{R}^n \rightarrow \mathbb{R}^m$,
(l_p, ϵ)-adversary region, i.e., $\{\boldsymbol{x}' \mid \|\boldsymbol{x}' - \boldsymbol{x}_0\|_p \leq \epsilon\}, p = \infty$

(2) To Certify: $\forall \boldsymbol{x} \in \{\boldsymbol{x}' \mid \|\boldsymbol{x}' - \boldsymbol{x}_0\|_p \leq \epsilon\}, N_{s_0}(\boldsymbol{x}) - N_{s_1}(\boldsymbol{x}) > 0$
where s_0 is the correct output label and s_1 is another output result.

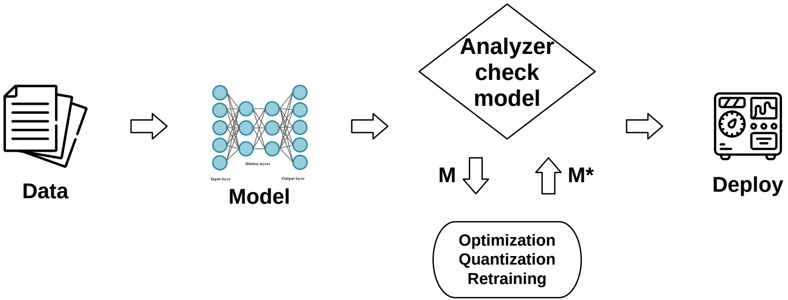
Previous Works

1. Scalable but imprecise
   - (approximate) Linear relaxation [ICML 2018]
   - (approximate) Abstract interpretation [S&P 2018, POPL 2019]

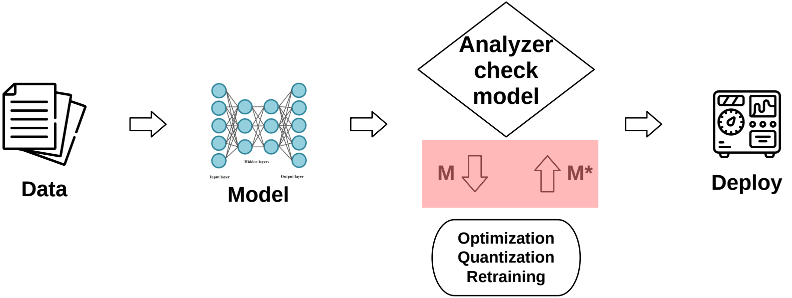
Previous Works

1. Scalable but imprecise
    - (approximate) Linear relaxation [ICML 2018]
    - (approximate) Abstract interpretation [S&P 2018, POPL 2019]
2. Precise but lacks scalability (time complexity is high, cannot handle large scale of network)
    - (exact verification) SMT solving [CAV 2017]
    - (adversarial attack) PGD crafts samples [CVPR 2017]
    - (metamorphic testing) neural coverage [ASE 2018]

Neural Network Deployment Workflow

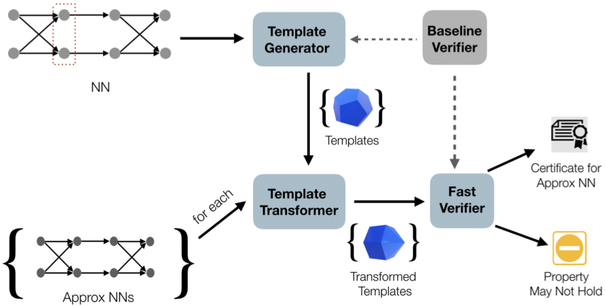


Neural Network Deployment Workflow



(3) Problem: The number of input cases in the adversary region is infinite: We cannot compute each $N(\boldsymbol{x}')$ for all separately.

OOPSLA2022:FANC

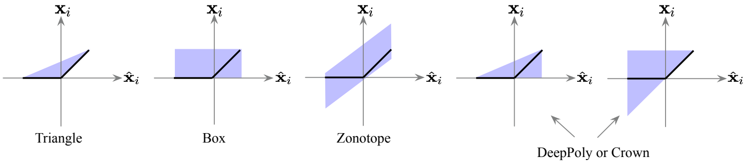


Abstraction Template for $\langle \Phi, \Psi \rangle$

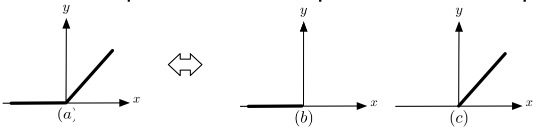
Branch and Bound for Verification

A widely used DNN complete verification technology.

- Bound: Efficient incomplete verification

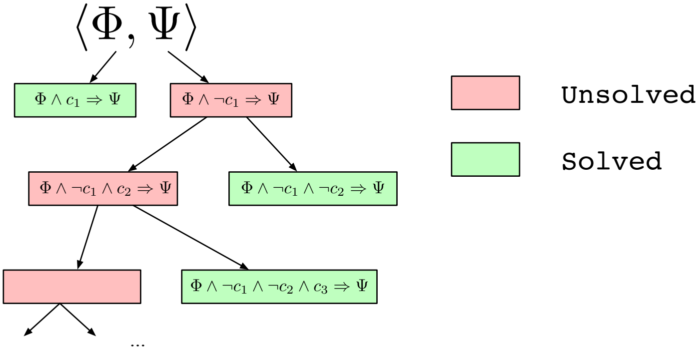


- Branch: Split verification problem into subproblem

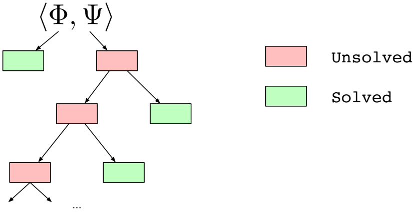


Branch and Bound for Verification

Challenge 1: How could we store the information and pass it on regarding bab?



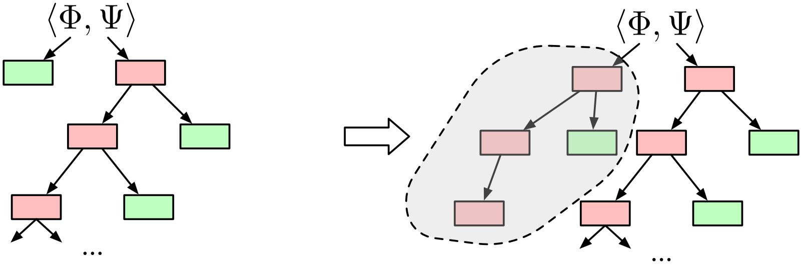
Technique 1: Reuse



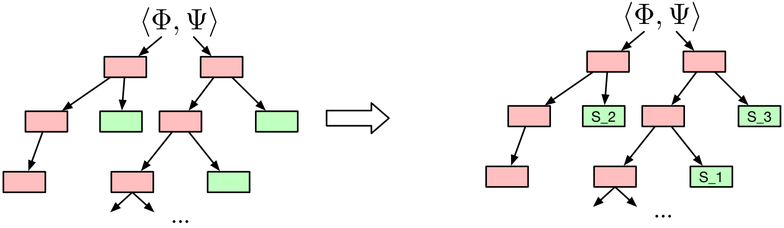
- Skip all unsolved problems.
- Start **Reuse** with all solved constraint on $\mathbf{M}*$

Guided by Counterexample

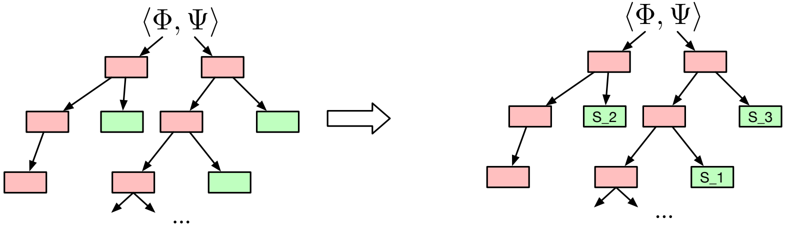
Challenge2: The **Reuse** template is not **enough** to verify **M**$*$.



Technique 2: Olive-G: Scoring Greedy Strategy



Technique 2: Olive-G: Scoring Greedy Strategy



- Update the constraint from each step.
- Scoring subproblem

Evaluation Setup

- **Baseline:**
  B_1: DeepZ
  B_2 Gurobi-based Lp verifier.
- **Verification Properties:**

Benchmark	Application	Neurons	Instances	Number
ACAS Xu **ACAS˙Xu**	Control Safety	300	$\{\mathbf{M_1} \dots \mathbf{M_{45}}\} \times \{p_1 \dots p_{10}\}$	186
RL **rl2022benchmarks**	Reinforcement Learning	128-512	$\{\mathbf{M_{46}} \dots \mathbf{M_{48}}\} \times \{p_{11} \dots p_{196}\}$	296
MNIST **muller2022third**	Computer Vision	512-1.5k	$\{\mathbf{M_{49}} \dots \mathbf{M_{52}}\} \times \{p_{197} \dots p_{230}\}$	90

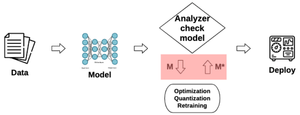
- **Modulate Networks**: Quantization to $INT8, INT16$; Retraining

Results

Model	Baseline	Similar Network	Our Tools
ACAS_Xu	B_1	$INT16$	**8.3X**
ACAS_Xu	B_2	$INT8$	**2.6X**
RL	B_1	$INT16$	**4.7X**
RL	B_2	$INT8$	**1.2X**
MNIST	B_1	$INT16$	**3.7X**
MNIST	B_2	$INT8$	**2.1X**

Summary

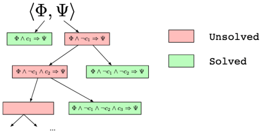
Neural Network Deployment Workflow



(3) Problem: The number of input cases in the adversary region is infinite:
We cannot compute each $N(\boldsymbol{x}')$ for all separately.

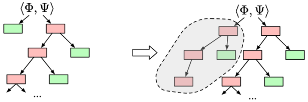
Branch and Bound for Verification

Challenge 1: How could we store the information and pass it on regarding bab?



Guided by Counterexample

Challenge2: The **Reuse** template is not **enough** to verify **M**∗.

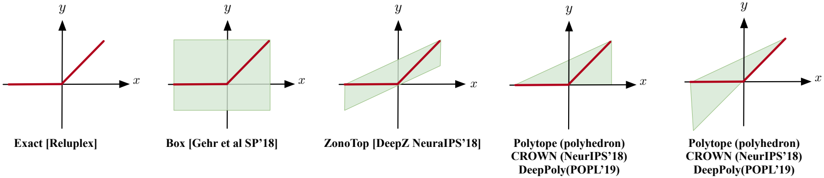


Results

Model	Baseline	Similar Network	Our Tools
ACAS_Xu	B_1	$INT16$	**8.3X**
ACAS_Xu	B_2	$INT8$	**2.6X**
RL	B_1	$INT16$	**4.7X**
RL	B_2	$INT8$	**1.2X**
MNIST	B_1	$INT16$	**3.7X**
MNIST	B_2	$INT8$	**2.1X**

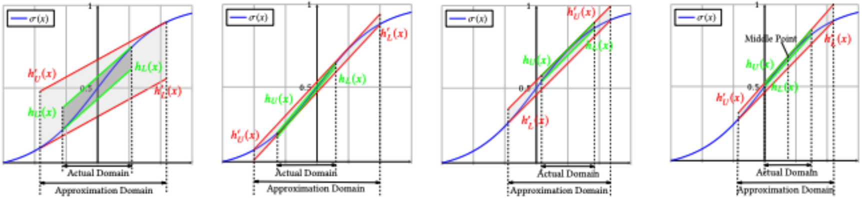
Quote: Deterministic

Complete Verification: A verifier is complete if it never reports false negatives (under all possible conditions).



Given enough time, the problem can be split and solved completely.

Quote: Deterministic



(a) Tangent line at end points [55]. (b) Minimal area [12]. (c) Parallel line [48, 53]. (d) Tangent line at middle point [12].

With any approximation approaches, they are always approximated and never solved by an exact approach, so they never achieve completeness.