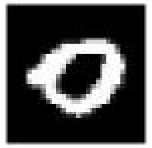
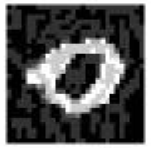
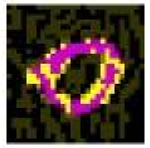
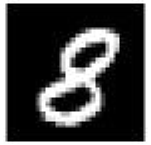
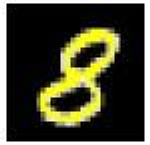
Robustness Certification of Neural Network Based on Abstract Interpretation

January 12, 2024

Kaijie Liu

Robustness Certification of Neural Network

- Robustness: consider a neural network that classifies an input x as having label y. Then, local robustness requires that all inputs x' that are "very similar" to x are also classified as having the same label y.

Attack	Original	Perturbed	Diff
FGSM [12], $\epsilon = 0.3$			
Brightening, $\delta = 0.085$			

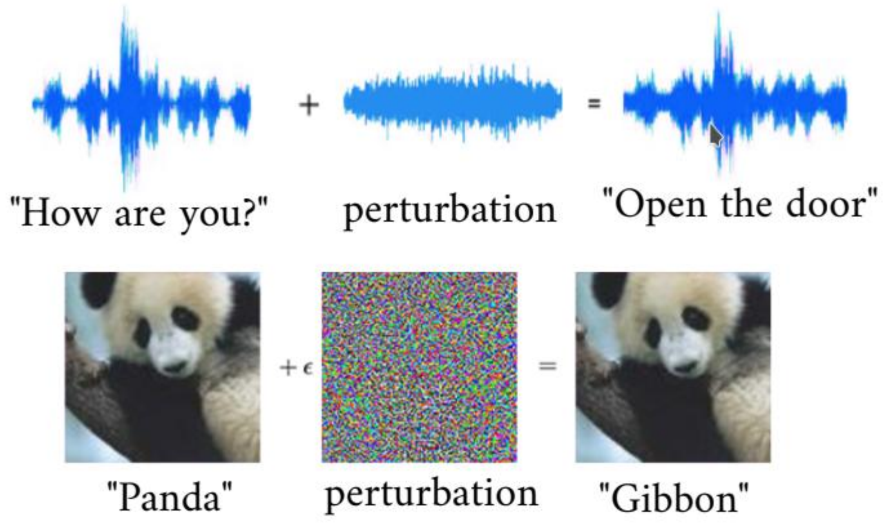
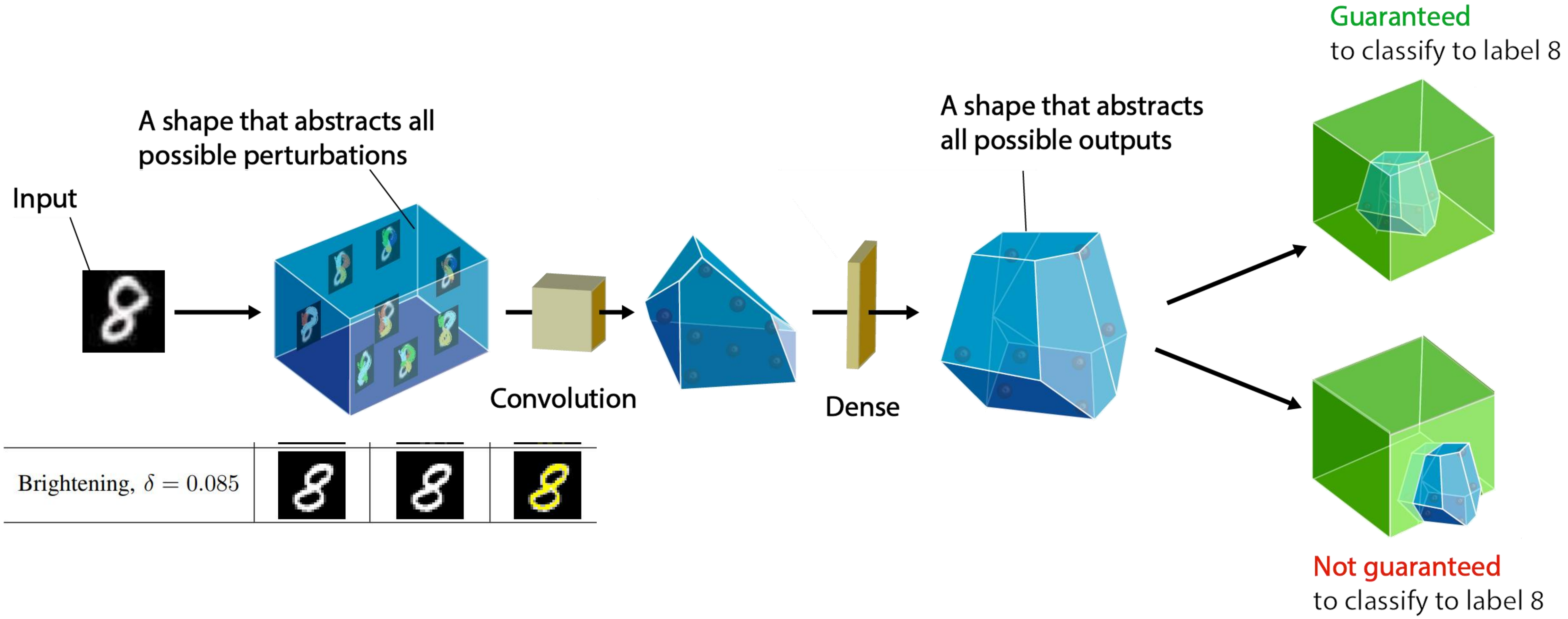


Figure 1. Small perturbations of the input cause the sound wave and the image to be misclassified.

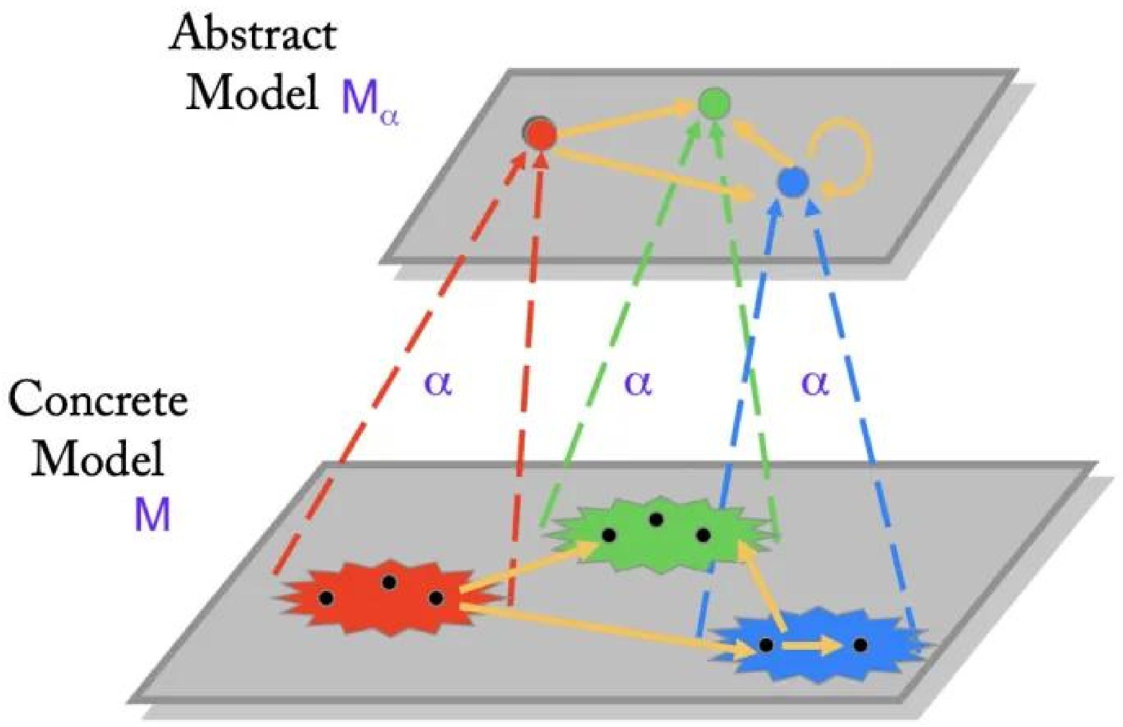
- The fast gradient sign method (FGSM)attack perturbs an image by adding to it a particular noise vector multiplied by a small number.
- The brightening attack perturbs an image by changing all pixels above the threshold $1 - \delta$ to the brightest possible value.

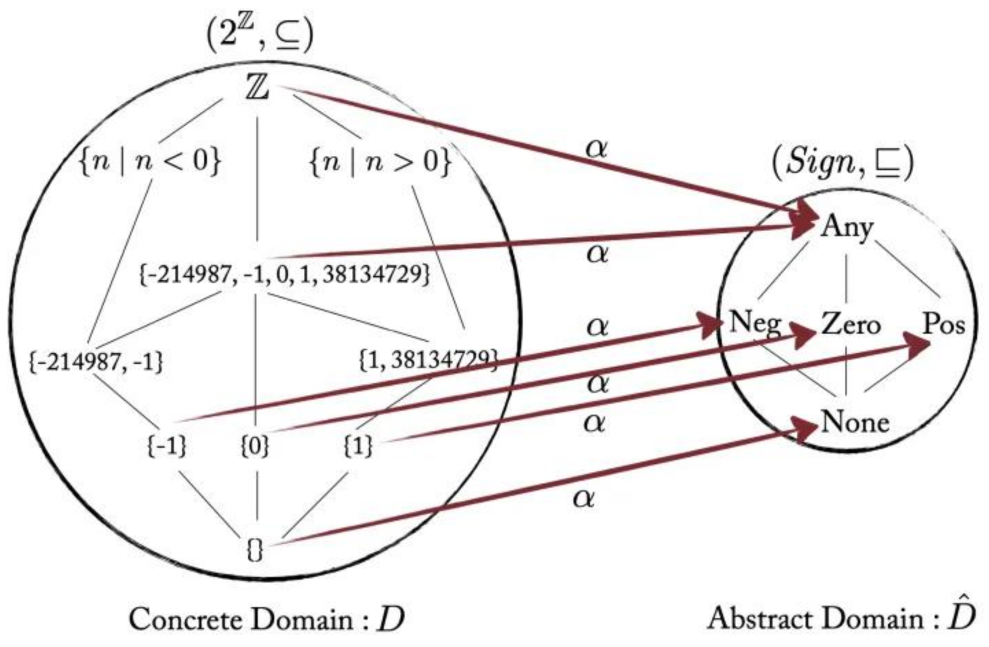
Robustness Certification of Neural Network Work Flow

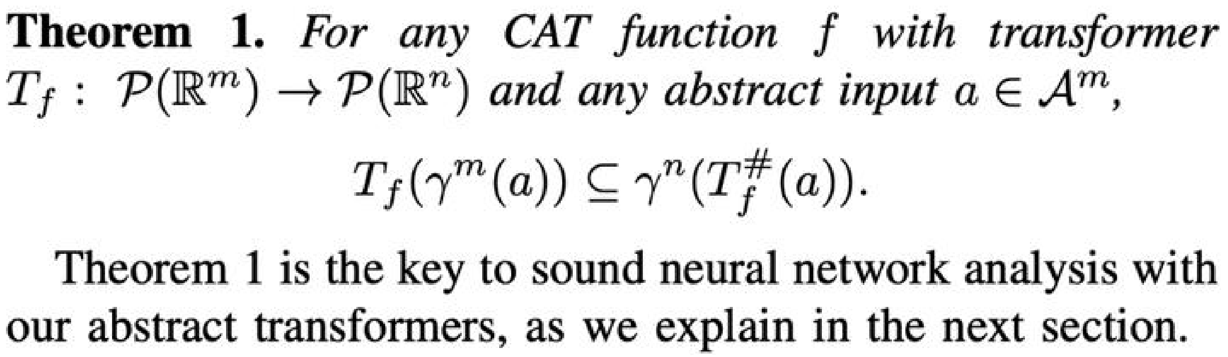


all perturbed images → Abstract element A1 → propagates A1 through abstract transformer → verifies

Abstract Interpretation







Theorem 1. *For any CAT function f with transformer $T_f : \mathcal{P}(\mathbb{R}^m) \to \mathcal{P}(\mathbb{R}^n)$ and any abstract input $a \in \mathcal{A}^m$,*

$$T_f(\gamma^m(a)) \subseteq \gamma^n(T_f^{\#}(a)).$$

Theorem 1 is the key to sound neural network analysis with our abstract transformers, as we explain in the next section.

Neural Network -- Formal Definition

- Neural network: $N_\theta: \mathbb{R}^d \rightarrow \mathbb{R}^k$,where *d* is input features, *k* is output classes, and parameterized by weights θ

$$N_\theta = class_i,\ i \in \{1,\ldots,k\},\ if\ N_\theta(x)_i > N_\theta(x)_j,\ j \neq i,\ x \in \mathbb{R}^d.$$

- Neural network Robust:

$$N_\theta(x) = class_i = N_\theta(\tilde{x})$$

$$\tilde{x} \in \pi(x), x \in \mathbb{R}^d$$

$$\pi: \mathbb{R}^d \rightarrow P(\mathbb{R}^d)$$

- Sound approximation for a given function:

$$f(\pi(x)) \subseteq A_{f,\pi}(x)$$

$$A_{f,\pi}: \mathbb{R}^d \rightarrow P(\mathbb{R}^k),\ f: \mathbb{R}^d \rightarrow \mathbb{R}^k, \pi: \mathbb{R}^d \rightarrow P(\mathbb{R}^d)$$

,sound approximations can be used to prove robustness properties.

How?

For example, if we can show that all values $\hat{z} \in A_{N,P_\in}(x)$ share the same $\mathrm{argmax}_i \hat{z}_i$

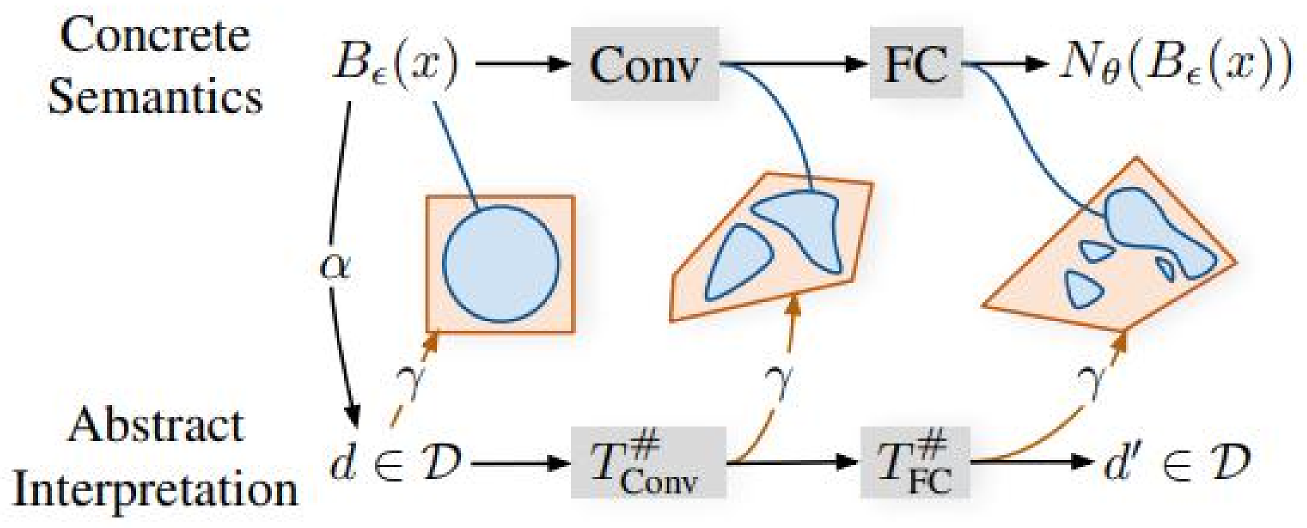
Neural Network Verification——Abstract Interpretation

- An abstract domain *D* is a set equipped with
  - an abstraction function: $\alpha : P(\mathbb{R}^p) \rightarrow D$
  - a concretization function: $\gamma : D \rightarrow P(\mathbb{R}^p)$
- An abstract transformer: $T_f^\# : D \rightarrow D'$ for a function: $f : \mathbb{R}^p \rightarrow \mathbb{R}^{p'}$

$$f(\gamma(d)) \subseteq \gamma'\left(T_f^\#(d)\right), d \in D$$

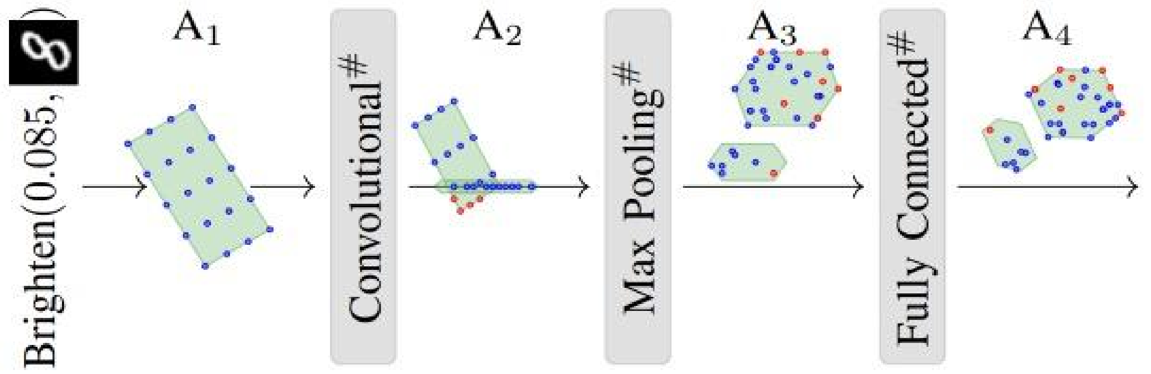
- Abstract transformers compose: If $T_f^\#$ and $T_g^\#$ are abstract transformers for functions *f* and *g*, then $T_f^\# \circ T_g^\#$ is an abstract transformer for $f \circ g$
- A sound approximation for *N* :

$$A_{N,\pi}(x) = \gamma\left(T_N^\#\left(\alpha(\pi(x))\right)\right)$$



AI2: Safety and Robustness Certification of NNs with Abstract Interpretation

CAT Functions. We express the neural network as a composition of conditional affine transformations (CAT), which are affine transformations guarded by logical constraints.



Any affine transformation $f(x) = W \cdot \overline{x} + \overline{b}$ is a CAT function, for a matrix ***W*** and a vector ***b***. Given sequences of conditions $E_1, \ldots, E_k$ and CAT functions $f_1, \ldots, f_k$:

$$f(\overline{x}) = caseE_1 : f_1(\overline{x}), \ldots, caseE_k : f_k(\overline{x})$$

Layers. Neural networks are often organized as a sequence of layers, such that the output of one layer is the input of the next layer. Layers consist of neurons, performing the same function but with different parameters. The output of a layer is formed by stacking the outputs of the neurons into a vector or three-dimensional array.

Reshaping of Inputs. A three-dimensional array

$$\overline{x} \in \mathbb{R}^{m \times n \times r} \rightarrow \overline{x}^v = \left(x_{1,1,1} \ldots x_{1,1,r} x_{1,2,1} \ldots x_{1,2,r \ldots} \ldots x_{m,n,1} \ldots x_{m,n,r}\right)^T$$

Activation Function. $\mathrm{Re}LU_i(\overline{x}) = \left(\mathrm{Re}LU(x_1), \ldots, \mathrm{Re}LU(x_m)\right)$

AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation

ReLU : $ReLU(\bar{x}) = (ReLU(x_1), \ldots, ReLU(x_n))$

$$ReLU = ReLU_n \circ ReLU_{n-1} \circ \ldots \circ ReLU_1$$

$$\begin{aligned} ReLU_i(\bar{x}) &= case(x_i \geq 0) : \bar{x}, \\ &= case(x_i < 0) : I_{i \leftarrow 0} \cdot \bar{x} \end{aligned}$$

Fully connected layer: $FC_{W,\bar{b}}(\bar{x}) = ReLU(W \cdot \bar{x} + \bar{b})$

Convolutional layer:

$$F^{p,q} = (F_1^{p,q}, \ldots, F_t^{p,q})$$

$$F_i^{p,q} : \mathbb{R}^{m \times n \times r} \rightarrow \mathbb{R}^{(m-p+1) \times (n-q+1)}$$

$$y_{i,j} = ReLU(\textstyle\sum_{i'=1}^{p} \sum_{j'=1}^{q} \sum_{k'=1}^{r} W_{i',j',k'} \cdot x_{(i+i'-1),(j+j'-1),k} + b)$$

$$\mathbb{R}^{m \times n \times r} \rightarrow \mathbb{R}^{(m-p+1) \times (n-q+1) \times t}$$

$$\Downarrow$$

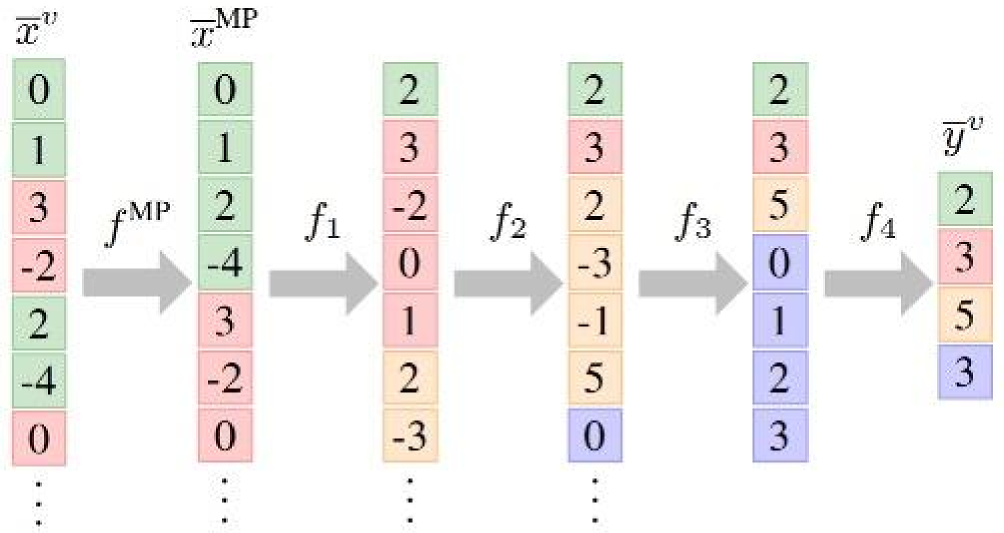
$$FC_{W^F,\bar{b}^F}(\bar{x}) = ReLU(W^F \cdot \bar{x}^v + \bar{b}^F)$$

MaxPool: $MaxPooling_{p,q} : \mathbb{R}^{m \times n \times r} \rightarrow \mathbb{R}^{\frac{m}{p} \times \frac{n}{q} \times r}$

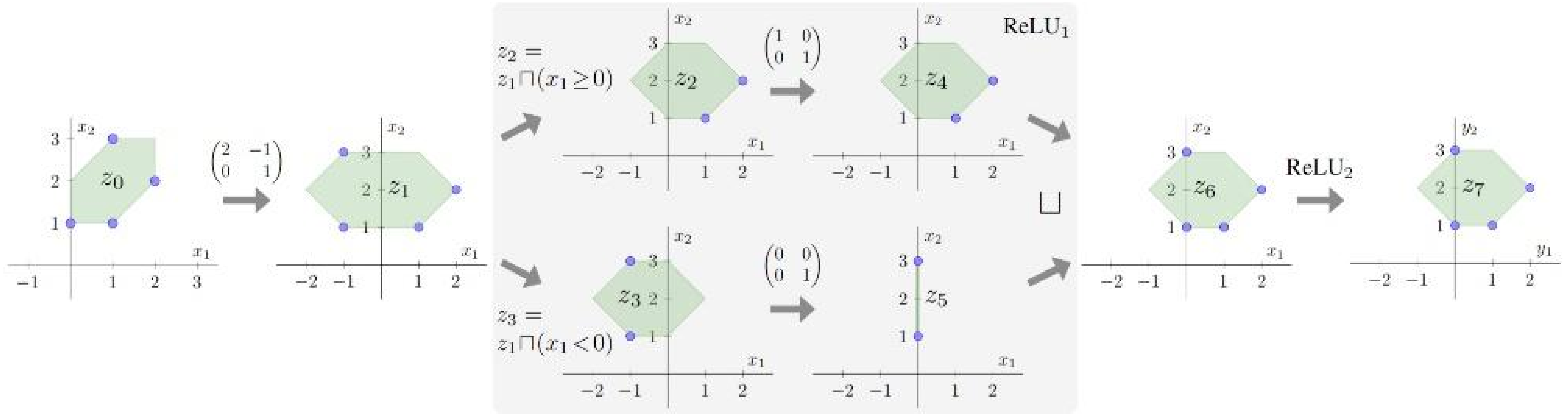
$$\begin{aligned} y_{i,j,k} = max(\{x_{i,j,k} \mid\ & p \cdot (i-1) < i' \leq p \cdot j \\ & q \cdot (j-1) < j' \leq q \cdot j\}) \end{aligned}$$

$$MaxPool'_{p,q} : \mathbb{R}^{m \cdot n \cdot r} \rightarrow \mathbb{R}^{\frac{m}{p} \cdot \frac{n}{q} \cdot r}$$

$$MaxPool'_{p,q} = f_{\frac{m}{p} \cdot \frac{n}{q} \cdot r} \circ \ldots \circ f_1 \circ f^{MP}$$



AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation



z_0
$\begin{pmatrix} 2 & -1 \\ 0 & 1 \end{pmatrix}$
z_1
$z_2 = z_1 \sqcap (x_1 \geq 0)$
z_2
$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
z_4
ReLU$_1$
$z_3 = z_1 \sqcap (x_1 < 0)$
z_3
$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$
z_5
$\sqcup$
z_6
ReLU$_2$
z_7

x_1
x_2

$$f(\bar{x}) = ReLU(W \cdot \bar{x})$$

$$W = \begin{pmatrix} 2 & -1 \\ 0 & 1 \end{pmatrix}$$

$$\begin{aligned} W \cdot (x_1, x_2)^T &= (2 \cdot (1 + 0.5 \cdot \epsilon_1 + 0.5 \cdot \epsilon_2) - (2 + 0.5 \cdot \epsilon_1 + 0.5\epsilon_3)) \\ &= (0.5 \cdot \epsilon_1 + \epsilon_2 - 0.5 \cdot \epsilon_3, 2 + 0.5 \cdot \epsilon_1 + 0.5 \cdot \epsilon_3) \end{aligned}$$

$$ReLU = ReLU_2 \circ ReLU_1 \qquad \begin{aligned} ReLU_i(\bar{x}) &= case(x_i \geq 0) : \bar{x}, \\ &= case(x_i < 0) : I_{i \leftarrow 0} \cdot \bar{x} \end{aligned}$$

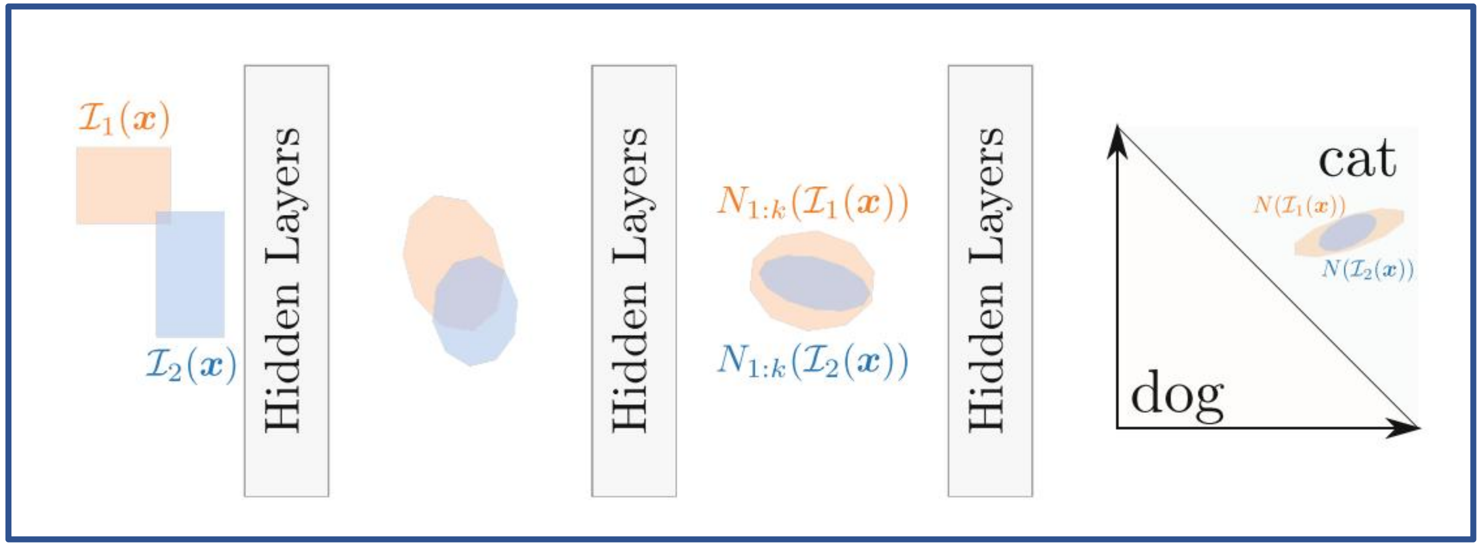
For $f(\bar{x}) = W \cdot \bar{x} + \bar{b}$, $T_f^{\#}(a) = \text{Aff}(a, W, \bar{b})$.

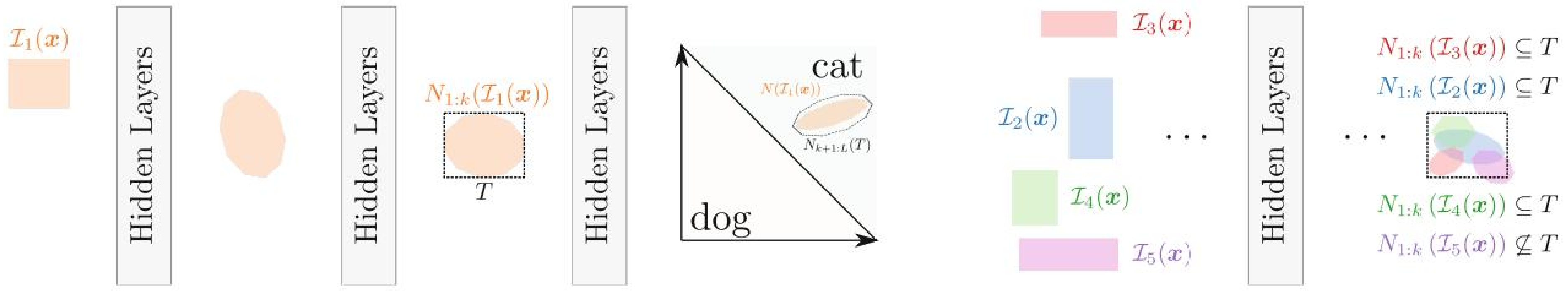
For $f(\bar{x}) = \textbf{case } E_1 : f_1(\bar{x}), \ldots, \textbf{case } E_k : f_k(\bar{y})$,

$$T_f^{\#}(a) = \bigsqcup_{1 \leq i \leq k} f_i^{\#}(a \sqcap E_i).$$

For $f(\bar{x}) = f_2(f_1(\bar{x}))$, $T_f^{\#}(a) = T_{f_2}^{\#}(T_{f_1}^{\#}(a))$.

Shared Certificates for Neural Network Verification -- CAV 2022


$\mathcal{I}_1(\boldsymbol{x})$
$\mathcal{I}_2(\boldsymbol{x})$
Hidden Layers
Hidden Layers
$N_{1:k}(\mathcal{I}_1(\boldsymbol{x}))$
$N_{1:k}(\mathcal{I}_2(\boldsymbol{x}))$
Hidden Layers
cat
$N(\mathcal{I}_1(\boldsymbol{x}))$
$N(\mathcal{I}_2(\boldsymbol{x}))$
dog


$\mathcal{I}_1(\boldsymbol{x})$
Hidden Layers
Hidden Layers
$N_{1:k}(\mathcal{I}_1(\boldsymbol{x}))$
T
Hidden Layers
cat
$N(\mathcal{I}_1(\boldsymbol{x}))$
$N_{k+1:L}(T)$
dog
$\mathcal{I}_3(\boldsymbol{x})$
$\mathcal{I}_2(\boldsymbol{x})$
$\mathcal{I}_4(\boldsymbol{x})$
$\mathcal{I}_5(\boldsymbol{x})$
Hidden Layers
$N_{1:k}(\mathcal{I}_3(\boldsymbol{x})) \subseteq T$
$N_{1:k}(\mathcal{I}_2(\boldsymbol{x})) \subseteq T$
$N_{1:k}(\mathcal{I}_4(\boldsymbol{x})) \subseteq T$
$N_{1:k}(\mathcal{I}_5(\boldsymbol{x})) \not\subseteq T$

(a) We generate a (verifiable) template T from the abstraction obtained by propagating the orange input.

(b) We shortcut the verification if intermediate abstraction are contained in the T.

DeepPoly: An Abstract Domain for Certifying Neural Networks

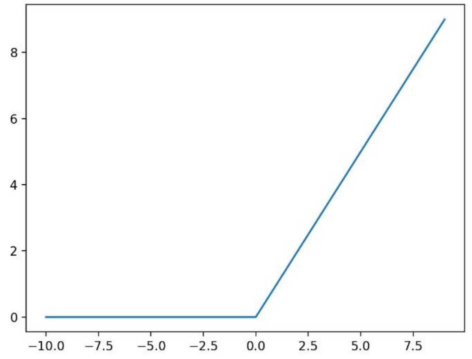
Attack	Original	Lower	Upper
L_∞			
Rotation			

Formally, an abstract element $a \in \mathcal{A}_n$ over n variables can be written as a tuple $a = \langle a^{\leq}, a^{\geq}, l, u \rangle$ where

$$a_i^{\leq}, a_i^{\geq} \in \{x \mapsto v + \textstyle\sum_{j \in [i-1]} w_j \cdot x_j \mid v \in \mathbb{R} \cup \{-\infty, +\infty\}, w \in \mathbb{R}^{i-1}\} \text{ for } i \in [n]$$

and $l, u \in (\mathbb{R} \cup \{-\infty, +\infty\})^n$. Here, we use the notation $[n] := \{1, 2, \dots, n\}$. The concretization function $\gamma_n \colon \mathcal{A}_n \to \mathcal{P}(\mathbb{R}^n)$ is then given by

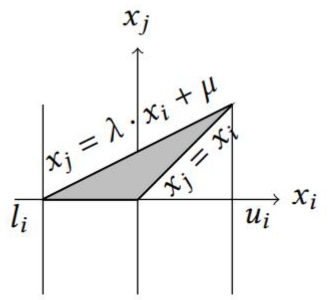
$$\gamma_n(a) = \{x \in \mathbb{R}^n \mid \forall i \in [n].\, a_i^{\leq}(x) \leq x_i \wedge a_i^{\geq}(x) \geq x_i\}.$$

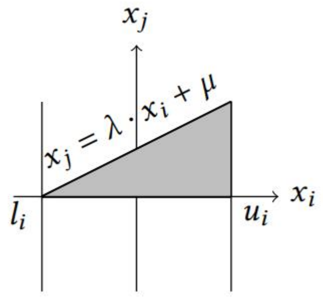


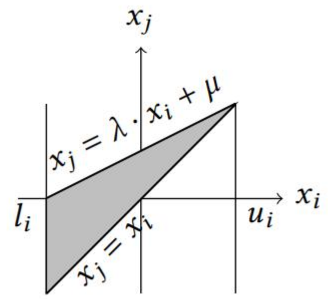
Let $f \colon \mathbb{R}^{i-1} \to \mathbb{R}^i$ be a function that executes $x_i \leftarrow v + \sum_{j \in [i-1]} w_j \cdot x_j$ for some $w \in \mathbb{R}^{i-1}$. The corresponding abstract affine transformer is $T_f^{\#}(\langle a^{\leq}, a^{\geq}, l, u \rangle) = \langle a'^{\leq}, a'^{\geq}, l', u' \rangle$ where $a_k'^{\leq} = a_k^{\leq}$, $a_k'^{\geq} = a_k^{\geq}$, $l_k' = l_k$ and $u_k' = u_k$ for $k < i$. Further, $a_i'^{\leq}(x) = a_i'^{\geq}(x) = v + \sum_{j \in [i-1]} w_j \cdot x_j$.

To compute l_i and u_i, we repeatedly substitute bounds for x_j into the constraint, until no further substitution is possible. Formally, if we want to obtain l_i', we start with $b_1(x) = a_i'^{\leq}(x)$. If we have $b_s(x) = v' + \sum_{j \in [k]} w_j' \cdot x_j$ for some $k \in [i-1]$, $v' \in \mathbb{R}$, $w' \in \mathbb{R}^k$, then

$$b_{s+1}(x) = v' + \sum_{j \in [k]} \left(\max(0, w_j') \cdot a_j'^{\leq}(x) + \min(w_j', 0) \cdot a_j'^{\geq}(x) \right).$$







Limitation & Future Work

- “Create” new abstract domain
- More tiny no-linear activation function OR Affine function
- More fast and scalable

Neural Network Verification Based on Abstract Interpretation

- AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation,
- Differentiable Abstract Interpretation for Provably Robust Neural Networks,
- Fast and Effective Robustness Certification,
- An Abstract Domain for Certifying Neural Networks,
- Boosting Robustness Certification of Neural Networks,
- Certifying Geometric Robustness of Neural Networks,
- Beyond the Single Neuron Convex Barrier for Neural Network Certification,
- Shared Certificates for Neural Network Verification

Robustness Certification of Neural Network Based on Abstract Interpretation

January 12, 2024

Kaijie Liu