Can certification methods benefit training?

Verifying networks which are not meant to be robust will certainly produce worse results (smaller epsilon provability) than verifying networks which are trained to be provably robust.

Note that there is a difference between training the network to be experimentally robust (e.g., PGD defense) vs. training the network to be provably robust (what we see next).

So, can we then use certification for training the network to be robust?
Recall: PGD Defense

find $\theta$ 
minimize $\rho(\theta)$
where $\rho(\theta) = \mathbb{E}_{(x,y) \sim D} \left[ \max_{x' \in S(x)} L(\theta, x', y) \right]$

$D$ is the underlying distribution

$\mathbb{E}$ is typically estimated with the empirical risk

$S(x)$ denotes the perturbation region around point $x$, that is, we want all points in $S(x)$ to classify the same as $x$. We can pick $S(x)$ to be:

$$S(x) = \{ x' | \|x - x'\|_\infty < \varepsilon \}$$

Madry et.al, 2017
Lets Incorporate Provability

\[
\begin{align*}
\text{find} & \quad \theta \\
\text{minimize} & \quad \rho(\theta) \\
\text{where} & \quad \rho(\theta) = \mathbb{E}_{(x,y) \sim D} \left[ \max_{z \in \gamma(NN^#(S(x)))} L(\theta, z, y) \right]
\end{align*}
\]

Differentiable Abstract Interpretation for Provably Robust Neural Networks
Mirman, Gehr, V. ICML 2018
Visualization of Certified Training

Essentially: **automatic differentiation of abstract interpretation**
**Adversarial Training**

\[
\begin{align*}
\text{find} & \quad \theta \\
\text{minimize} & \quad \rho(\theta) \\
\text{where} & \quad \rho(\theta) = \mathbb{E}_{(x,y) \sim D} \left[ \max_{x' \in S(x)} L(\theta, x', y) \right]
\end{align*}
\]

Find input $x'$ that achieves high loss

**Certified Defense**

\[
\begin{align*}
\text{find} & \quad \theta \\
\text{minimize} & \quad \rho(\theta) \\
\text{where} & \quad \rho(\theta) = \mathbb{E}_{(x,y) \sim D} \left[ \max_{z \in \gamma(NN^#(S(x)))} L(\theta, z, y) \right]
\end{align*}
\]

Find output $z$ that achieves high loss (under abstraction)
Certified Defenses: General Method

\[ \max_{z \in \gamma(\text{NN}^\#(S(x)))} L(\theta, z, y) \]

Let us examine the pattern in the concrete first.
The pattern works with any abstract relaxation.

Differentiable Abstract Interpretation for Provably Robust Neural Networks
Mirman, Gehr, V. ICML 2018
Let us now pick a loss function $L$

$$L(z, y) = \max_{q \neq y} (z_q - z_y)$$
Certified Defenses with a given loss

\[
\max_{z \in \gamma(NN^\#(S(x)))} L(\theta, z, y)
\]

Let's define \( L \) to be:

\[
L(z, y) = \max_{q \neq y} (z_q - z_y)
\]

\( \gamma \) captures possible perturbations, e.g., \( L_\infty \) ball

\( z \in \gamma(NN^\#(S(x))) \)

Differentiable Abstract Interpretation for Provably Robust Neural Networks
Mirman, Gehr, V. ICML 2018
Certified Defenses with a given loss

\[
\max_{z \in \gamma(NN \#(S(x)))} L(\theta, z, y)
\]

Key problem: set of vectors could be infinite or very large, so we cannot just enumerate.

How do we address this?

Let's define \( L \) to be:

\[
L(z, y) = \max_{q \neq y} (z_q - z_y)
\]
Certified Defenses in the abstract

\[
\max_{z \in \gamma(\text{NN}^\#(S(x)))} L(\theta, z, y)
\]

Let's define \( L \) to be:

\[
L(z, y) = \max_{q \neq y} (z_q - z_y)
\]

e.g. \([0.1, 0.3]\]

This is an affine transform which is exact for zonotope.

\[d_0 = z_0 - z_y\]

\[d_9 = z_9 - z_y\]

\[\max(\text{box}(d_0))\]

\[\max(\text{box}(d_9))\]

Plug in lower and upper bounds of epsilons into the expression to get the interval.
Defining \( \text{max}(\text{box}(d_0)) \)

\[
d_0 = 3 + \epsilon_1 - 2\epsilon_2
\]

\( \epsilon_1 \) and \( \epsilon_2 \) range over \([-1,1]\)

\[
d_{-1,-1} = 3 - 1 + 2 = 4
\]

plug in \(-1\) for both

\[
d_{-1,1} = 3 - 1 - 2 = 0
\]

plug in \(-1\) for \(\epsilon_1\), and \(1\) for \(\epsilon_2\)

\[
d_{1,-1} = 3 + 1 + 2 = 6
\]

plug in \(1\) for \(\epsilon_1\), and \(-1\) for \(\epsilon_2\)

\[
d_{1,1} = 3 + 1 - 2 = 2
\]

plug in \(1\) for both

\[
d_{\text{box}} = [0,6] \quad \text{max}(d_{\text{box}}) = 6
\]

Of course, to compute max, rather than enumerating combinations, we pick the value for the \(\epsilon\) depending on its sign in \(d_0\). If positive, pick 1, if negative, pick -1.
Let us keep the same pattern but now pick a different loss, the cross-entropy loss $CE$

$$L(z, y) = CE(z, y)$$

This is in the concrete, but we need to work in the abstract.

On the Effectiveness of Interval Bound Propagation for Training Verifiably Robust Models, 2018
Gowal, Dvijotham, Stanforth, Bunel, Qin, Uesato, Arandjelovic, Mann, Kohli
Let us keep the same pattern but now pick a different loss, the **cross-entropy loss** $CE$

\[ L(z, y) = CE(z, y) \]
Few additional tricks in practice

- Annealing on the size of $S(x)$ – start with small region around $x$ (small $\epsilon$) and gradually grow it during training. This was found to be most helpful heuristic.

- Even though the whole propagation is done via Box, IBP processes the last linear layer exactly (e.g., zonotope).

- Dynamically weighing-in the standard CE loss and the correctness CE loss.

These and more implemented in the DiffAI certified training system: https://github.com/eth-sri/diffai
Key observations when using DiffAI scheme in practice:

Using cheap relaxations (e.g., Box) scales to large networks. But the problem is, it introduces a lot of garbage (infeasible points) in the final output shape, meaning the deeper the network is, the more the capacity increases (potential for higher accuracy), but the more we are training w.r.t. assigning labels to garbage points. Thus, typically training with Box scales but accuracy drops substantially.

Naturally we would like to reduce the infeasible points w.r.t to which we are training. However, it turned out that more precise relaxations (e.g., Zonotope) may lead to worse results than Box! This is an unintuitive pathological situation where more precise relaxations during training do not actually bring better results in provability and where further loss tweaking is not enough.
Question I:

Is there a network with perfect accuracy s.t. analyzing it with Box is exact?
Question I:
Is there a network with \textbf{perfect accuracy} s.t. analyzing it with Box is \textbf{exact}?

Yes.

Universal Approximation with Certified Networks, ICLR’2020
Baader, Mirman, Vechev

(answers the existence question, but construction is still impractical, more work needed)
Question II:

Why are better relaxations not producing better results?
Question II:

Why are better relaxations not producing better results?

Hypothesis: More complex abstractions lead to more difficult optimization problems.

Why? Intuitively, a relatively small number of weights in the network need to control complex relaxations with many more parameters (than weights). This is quite unlike normal training.
Question II:

Why are better relaxations not producing better results?

We need a training method that produces a \textit{simpler optimization problem}
Reminder: optimization problems

Adversarial Training

\[ \text{find} \ \theta \quad \text{minimize} \ \rho(\theta) \]
\[ \text{where} \quad \rho(\theta) = \mathbb{E}_{(x, y) \sim D} \left[ \max_{x' \in S(x)} L(\theta, x', y) \right] \]

Find input \( x' \) that achieves high loss

- Good accuracy
- Worse verifiability
- Easier optimization

Certified Defense

\[ \text{find} \ \theta \quad \text{minimize} \ \rho(\theta) \]
\[ \text{where} \quad \rho(\theta) = \mathbb{E}_{(x, y) \sim D} \left[ \max_{z \in NN''(S(x))} L(\theta, z, y) \right] \]

Find output \( z \) that achieves high loss (under abstraction)

- Worse accuracy
- Good verifiability
- Harder optimization
Adversarial Training and Provable Defenses: Bridging the Gap

COLT: Balunovic and V, ICLR’20 (oral)

Key challenge:

Find point $x_1 \in S_1$ such that loss $L$ in the final layer is maximized

Need projections again!

Optimization problem (after layer 1):

$$\min_{\theta} \max_{x_1 \in S_1} L(h_3(h_2(x_1)), y_{true})$$

(high-level view: PGD training but with shapes arising in the middle of the network)

https://github.com/eth-sri/colt
Instantiation of COLT with Zonotope

\[
\max_{e \in [0,1]^k} L(Ae, y_{true}) = \max_{x \in Z} L(x, y_{true})
\]

Zonotope relaxation (reminder): \( Z = A \cdot [-1,1]^k \)

Each \( x \in Z \) has a corresponding \( e \in [-1,1]^k \) such that \( x = Ae \)

Solved via standard PGD (projection on Box)

Projection on Zonotope solved via projection on Box (via change of variables)
Lecture Summary

Certified defenses: using relaxations during training in order to obtain more provable networks

We introduced the DiffAI method and showed how to instantiate it with two loss functions and two relaxations

The DiffAI method and its follow-ups can produce complex optimization problems. Towards that, we introduced COLT, a certified defense that combines adversarial training and relaxations to produce a simpler optimization problem where better relaxations can lead to better results.