Probabilistic Verification of Neural Networks

Saksham Bhatnagar

Independent

June 17, 2020

Approach 1: Probabilistic Robustness (Overview)

- Main Idea: To provide a probabilistic guarantee for the effectiveness of a neural network over a closely related set of inputs
- The Problem:
  - Given a single x' the set of closely related (δ-close) inputs x are defined as $||x' - x|| \leqslant \delta$
  - Robustness requires same output for close inputs i.e. $\forall x. ||x' - x|| \leqslant \delta \implies f(x') = f(x)$
  - Extending this measure of robustness to all inputs is expensive and too strong a requirement

Approach 1: Probabilistic Robustness (Solution)

- Solution:
  - Create a probabilistic measure of robustness that guarantees correctness over a distribution of inputs.
  - This measure is defined as $Pr(||f(x') - f(x)|| \leqslant k * ||x' - x|| \mid ||x' - x|| \leqslant \delta) \geqslant 1 - \epsilon$, where $x' \sim D$
  - Hence for a given distribution D and error level ϵ if the probability that the output is more different than inputs exceeds ϵ then the neural network is not robust.

Approach 1: Probabilistic Robustness (Process)

- The Process:
  - The procedure creates an abstract representation of the neural network.
  - The abstract representation is used to create adversarial input sets.
  - A Monte Carlo algorithm (Importance Sampling) is used to generate examples from each input set.
  - The error is calculated on each input set as proportion of adversarial samples classified as non-adversarial.
  - If this error is greater than ϵ then the neural network is not robust.

Approach 2: Confidence Region (Overview)

- Main Idea: Given a 'safe' region for the output find the probability output remains in the region for perturbed input. Further given a confidence region for input estimate confidence region for output
- The Problem:
  - Probabilistic Verification: The problem is to find the lower bound for the probability that output remains in the same region. This translates to the following optimization problem, which is a non-convex optimization problem.

$$\text{maximize } p \text{ subject to } f(\varepsilon^p) \subseteq S \text{ and } p \in [0, 1)$$

where $f(\varepsilon^p)$ is the output and S is the 'safe' region.

Approach 2: Confidence Region (Overview contd.)

- The Problem (contd.):
  - Confidence Propagation: The problem is to find the smallest confidence region for output given the input is derived from an ellipsoid ε^p for a probability level p. It can be written as the following optimization problem.

$$\text{minimize } Volume(S) \text{ subject to } f(\varepsilon^p) \subseteq S$$

where $f(\varepsilon^p)$ is the output and S is the 'safe' region.

Approach 2: Confidence Region (Solution)

- The Solution:
  - Create an abstract representation of the activation function (ReLU, tanh, softmax etc.). This is in the form of quadratic constraint on an optimization problem.
  - Formulate the neural the network as constraints to the volume minimization problem using three matrices M^1, M^2, and M^3.
  - The constraints describe the input as an ellipsoid bounded region (M^1), the activation function (M^2), and 'safe' region of the neural network's output (M^3).The problem is a semi-definite programming problem.
  - The solution to optimization yields an ellipsoid that bounds the output $f(\varepsilon^p)$. This set is enclosed by an ellipsoid ε^p

Approach 2: Confidence Region (Process)

- The Process:
  - Select an input set X such the $Pr(X \in \varepsilon^p) \geqslant p$. The ellipsoid ε^p is given by
    $$\varepsilon^p = (x - \mu)^T \sum^{-1}(x - \mu) \leqslant n/(1 - p)$$
    $$\text{where } X \in R^n \text{ , } E[X] = \mu \text{ and } Cov[X] = \sum$$
    This bounded region can be expressed a quadratic constraint (M^1)
  - The activation function is encoded as a quadratic constraint using a symmetric indefinite matrix Q and is defined more explicitly in M^2
  - Finally the 'safe' region for the output is also encoded as a quadratic constraint in M^3.
  - It can then be shown if the constraint $M^1 + M^2 + M^3 \leq 0$ holds then the output $y \in \varepsilon(\mu^y, \sum^y)$

Propositions

1. Improve the sampling procedure by using different MC algorithms in probabilistic robustness to improve the quality of the robustness measure.
2. Extend probabilistic robustness to neural networks producing categorical outputs using softmax outputs.
3. The confidence region method is an extension of a generic approach. The method can be extended to check the performance of neural networks on non-elliptical distributions of input.
4. Combine the two approaches that by using input abstraction methods in confidence region approach as abstract interpreters in probabilistic robustness.

Bibliography

- Mangal, R. et.al Robustness of Neural Networks: A Probabilistic and Practical Approach. Retrieved June 09, 2020, from https://arxiv.org/abs/1902.05983
- Fazylab, M. et.al Probabilistic Verification and Reachability Analysis of Neural Networks via Semidefinite Programming. Retrieved June 09, 2020, from https://arxiv.org/abs/1910.04249
- Fazylab, M. et.al Safety Verification and Robustness Analysis of Neural Networks via Quadratic Constraint and Semidefinite Programming. Retrieved June 10, 2020, from https://arxiv.org/abs/1903.01287