RIAI 2019 Course Project

Samriddhi Jain (samjain)
Kumar Mohit (kmohit)

December 19, 2019

1 Objective

We aim to implement a sound verifier to certify the robustness of the given trained neural network models against adversarial attacks using DeepZ relaxation on the pixel bounds.

2 Implementation of Zonotope

Below are some key points involved in our implementation,

- Initial Zonotope: We assume one independent error term for each pixel, and use the given epsilon as its coefficient. Next to make sure the bounds for each pixel are within [0,1], we clip the zonotope in concerned dimensions, shift the center and update the epsilon accordingly. Since the first layer of each zonotope is normalization, we apply the same to the error coefficients as well.
- FC and Conv Layers: Propagation through FC and Conv layer is simply the forward pass through the layers, with coefficients of each error term being treated as a separate batch input.
- Propagation through Relu: For propagating through relu, we first compute the optimum lambda. We implement the two set of zonotopes as suggested in [1] and propagate it according to its value compared to the optimum lambda.
- Verification: For the final verification, rather than comparing the upper bound of each class with the lower bound of true label, we first subtract the zonotope coefficients of the true label from the zonotopes of remaining classes and verify that the upper bound of all are negative. The reasoning behind this is taking advantage of the epsilon terms, which can get cancelled out with subtraction, leading to more precise bounds.

3 Learning Lambda

For efficient verification we employ the idea of gradient descent, inspired by this work [2]. We create a tensor which holds lambda value for each Relu node. We propagate our input zonotope along with the lambda values, create a loss from the final verification bounds and then update the lambda values by computing the gradient with respect to the lambda. We construct the cost function as maximum positive value after the verification step. It essentially represents the maximum deviation of incorrect class with respect to true label and we minimize that. We use Adam optimizer with ReduceLROnPlateau LR scheduler. The cost function, learning rates and other parameters were empirically determined.

4 Test Suite

In order to affirm the soundness of our implemented model, we used PGD attack on several images of the MNIST test dataset to generate new test images. The epsilon value corresponding to the L-inf ball around each pixel was kept small (0.005-0.2) to ensure that the generated images correspond closely to the test cases on which the model is to be tested.

References

[1] Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, Martin Vechev. Fast and Effective Robustness Certification.

[2] Learning to learn by gradient descent by gradient descent.