Project proposal - extension of "Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks"

Szymon Gołębiowski Alex Lalov

The method presented in the Reluplex paper [1] is able to verify neural networks that consist of linear layers and non-linear ReLU activation functions. We want to explore if Reluplex could work with Convolutional Neural Networks on image data. Specifically, we want to use Reluplex to verify to what degree an image can be noised before a CNN outputs the wrong classification. In addition to layers already handled by Reluplex, CNNs include convolutions and max pooling layers.

Our main priority is to make Reluplex compatible with convolutional layers. The convolutions will be presented to the Reluplex engine in a similar fashion to a multi-layer perceptron by flattening the input space.

A further extension that we will attempt to evaluate is implementing max pooling in the Reluplex engine as an activation function similar to ReLU. The max function is relatively difficult to implement as the engine cannot handle a function with multiple parameters but theoretically possible because the function is piecewise linear.

In terms of the output of a supervised learning classification problem, we will simply retrieve the maximum value among the output layer's neurons without any further analysis or application of a softmax function because it is monotonic.

We posit that the current implementation of Reluplex will not be efficient enough to handle the large input state space of a square 480x480p image, as it includes ∼230,400 input values. Hence, we will run the evaluation on a low-resolution one-dimensional dataset MNIST-1D [2] that was created for the purpose of conducting experiments with computationally demanding pattern recognition methods (examples in Figure 1).

Lastly, if we're able to verify a network which includes both max pooling and learned kernels, we should be able to verify the entire convolutional neural network and determine the lower and upper bounds of a given image's pixel inputs. Furthermore we should be able to determine just how much noise can be added before the image is misclassified as we will be able to average the lower and upper bounds of pixels.

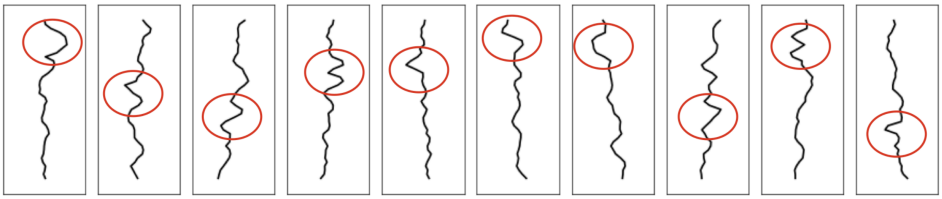

Figure 1: One-dimensional samples from the MNIST-1D dataset. The red circles highlight the patterns that determine the assignment to one of the ten classes.

References

[1] Guy Katz et al. *Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks*. 2017. arXiv: 1702.01135 [cs.AI]. URL: https://arxiv.org/abs/1702.01135.

[2] Sam Greydanus and Dmitry Kobak. *Scaling Down Deep Learning with MNIST-1D*. 2024. arXiv: 2011.14439 [cs.LG]. URL: https://arxiv.org/abs/2011.14439.