1 Heuristics

Neuronwise Heuristic Produce a scoring mechanism that scores a neuron based on some importance criteria and thus chooses the best neurons on which to perform linear programming per layer.

Weight Scores Heuristic This heuristic looks at a neuron's outgoing weights and the neurons's bounds to determine its score. The score is obtained by multiplying the neuron's outgoing weights times either its upper bound or the lower bound or the absolute difference between these. The scoring policy is quite slow in the current implementation.

Moving Window Linear Programming Heuristic This heuristic, as the name suggests, consists on performing linear programming on an arbitrary window of layers of the network. This partial model is then moved across the network as if it were a window. This heuristic verifies with very high precision but takes fractions of the time of the full layerwise linear programming.

Recursive Back-Prop of High Impact Neurons Firstly, fast interval propagation is performed to have a general idea of the intervals each neuron inside the network can take. Then, starting at the output layer:

1. For each neuron n in the layer l, check which neurons in the previous layer $l-1$ can affect its value the most. This is performed by checking the possible interval size of each incoming neuron m and multiplying it by the weight between m and n. This gives a general extimation how much m affect the output interval of n. The scoring policy can be one of the one explained in the Neuronwise heuristic.

2. Based on the scores computed in the previous step, take the highest capacity neurons that affect neuron n and store them.

3. Compute the union of all high impact sets returned.

4. Repeat from step 1 using the previous layer (layer that so far contained the m neurons), but only check for high impact neurons in the list returned from step 2.

The algorithm end up having a list of high impact neuron sets for each layer. Therefore, starting at the first hidden layer, linear programming is computed on the high impact neurons. For the non-high impact neurons within the current layer, the previously computed ELINA bounds are kept.

2 Final Remarks

For all networks except the mnist_relu_4_1024, the linear programming implementation runs fast enough on all epsilon values. The only exceptions to this rule are images that are not verifiable anyways for the given epsilon. For the mnist_relu_4_1024 network, the Recursive Back-Prop of High Impact Neurons heuristic was adopted. There the capacity used is dependent on the epsilon value of the perturbed image.