Abstraction based Output Range Analysis for Neural Networks

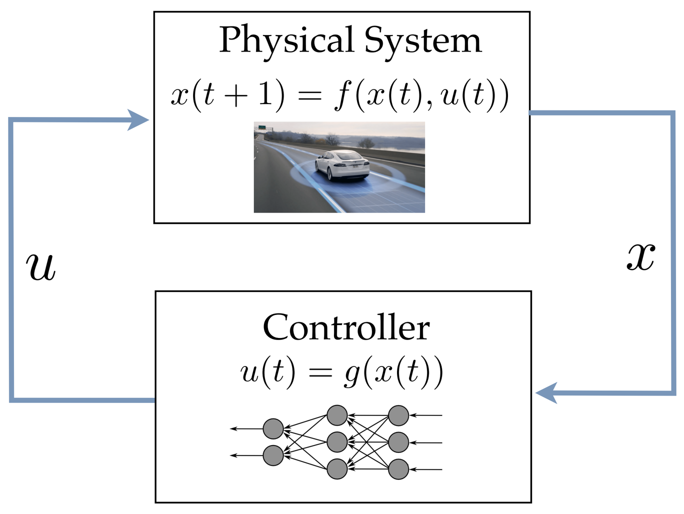
KANSAS STATE UNIVERSITY

Pavithra Prabhakar Zahra Rahimi Afzal

Department of Computer Science, Kansas State University, Manhattan, KS

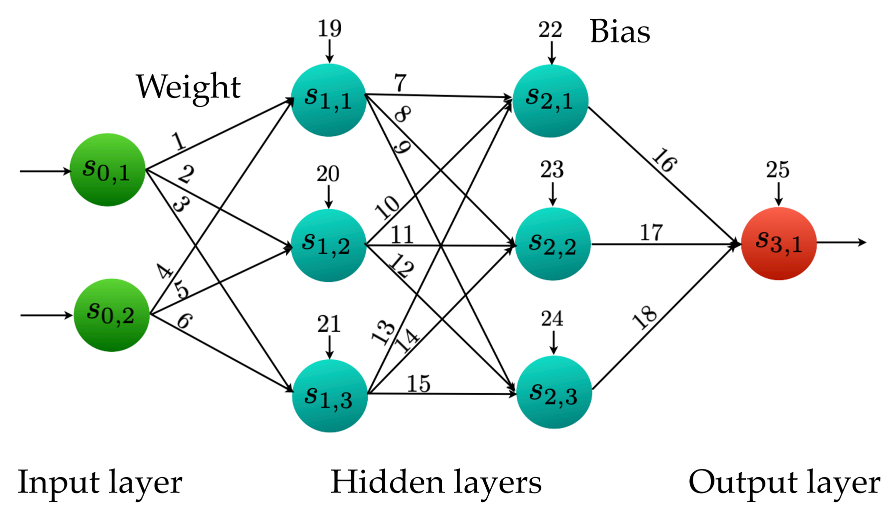
Motivation

- Classical control design methodologies fall short for the design of novel functionalities, such as, to provide autonomy in ground and aerial vehicles
- Traditional feedback controllers are replaced by learning based components such as artificial neural networks



- Neural network controlled physical systems operate in safety critical environments
- Need to provide rigorous guarantees on the functioning of these systems
- Safety is an important specification that stipulates that every execution of the systems is error free
- E.g. the autonomous vehicle always remains within the lane

Neural Network (NN)



- A neural network with 4 layers
  - An input layer, two hidden layers and an output layer
  - Weights on edges connecting consecutive layers, and biases on nodes
- Semantics captures input output valuations
  - The value at a node is obtained by the sum bias at the node and sum of the products of the weights and the values at the source of the incoming edges

$$V(s_{1,3}) = V(s_{0,1}) * 3 + V(s_{0,2}) * 6 + 21$$

Output Range Analysis Problem

- The crux of safety analysis lies in computing the reachable set, that is, the set of all outputs values given a set of input values

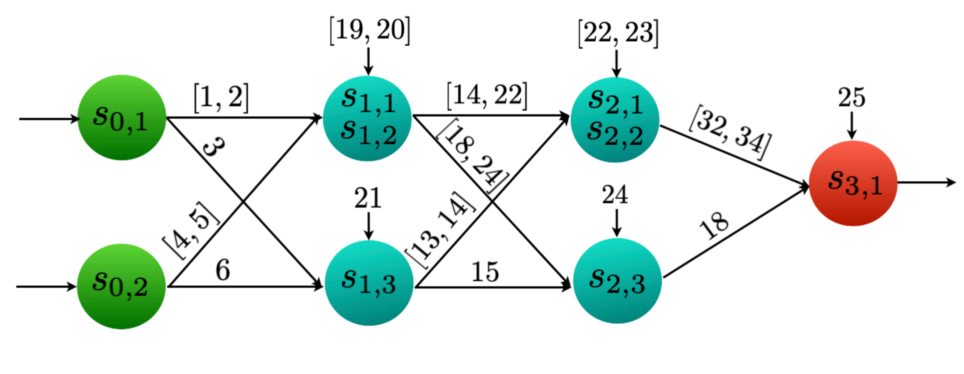
> Given a neural network $\mathcal{T}$, and a set of values I for the input layer, compute a range of values $[v_{min}, v_{max}]$ for the corresponding values of an output node.

- Current approaches:
  - MILP based encoding (Sherlock), satisfiability modulo solvers (Reluplex)
- Challenges:
  - Scalability with respect to the network size
  - MILP/SMT solving is expensive, and size of the constraints is proportional to the size fo the network

Abstraction based Analysis

- Abstraction: Construct a smaller "interval neural network" (INN) that over-approximates the behavior of a given network
- INN output range analysis: Extend the MILP encoding to compute the output range

Interval Neural Network (INN)



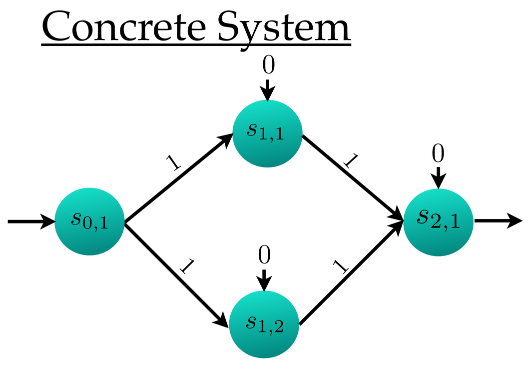
$$V(s_{1,1}, s_{1,2}) = V(s_{0,1}) * 1.5 + V(s_{0,2}) * 4.5 + 19.5$$

- Extends a neural network with "interval" weights and biases
- The value at a node is computed as before by choosing some value for weight and biases from their corresponding intervals

Step 1: Abstraction of NN to INN

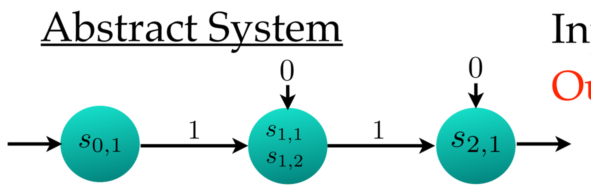
- Partition nodes of a layer and merge
- Approximate weights and bias by intervals
- First try: Take interval hull of the weights of edges (biases) being merged
- Fix: Scale the interval by a factor which is the number of nodes being merged in the source
- Can be interpreted as a left abstraction with scaling followed by a right abstraction without scaling

Concrete System



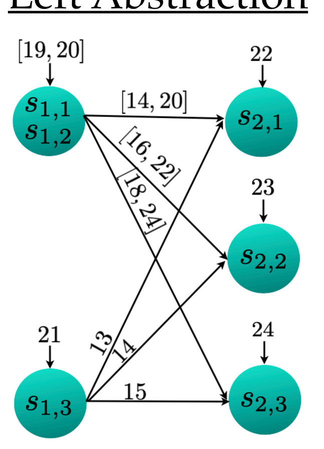
Input: 1
Output: 2

Abstract System

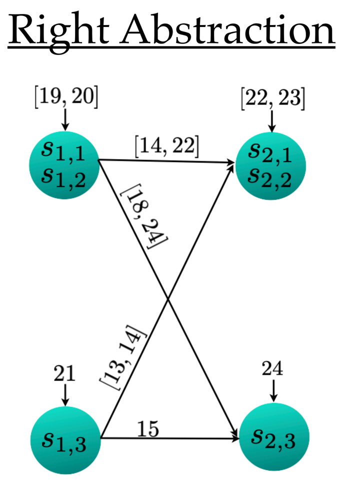


Input: 1
Output: 1

Left Abstraction

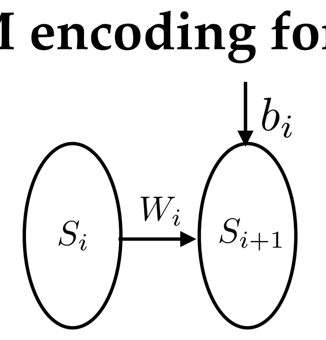


Right Abstraction



Step 2: Encoding of INN to MILP

Big-M encoding for NN



For $s' \in S_{I+1}$, constraints $C_{s'}^{I+1}$:

$$\sum_{s \in S_i} W_i(s, s')x_s + b_i(s') \leq x_{s'}$$
$$\sum_{s \in S_i} W_i(s, s')x_s + b_i(s') + Mq_{s'} \geq x_{s'}$$
$$0 \leq x_{s'},\ M(1 - q_{s'}) \geq x_{s'}$$

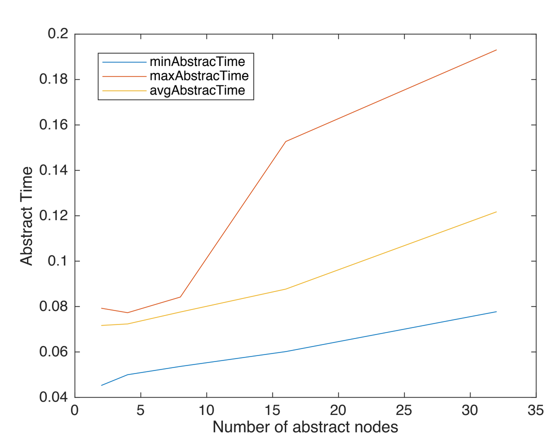
x_s represents the value at node s
q_s is a Boolean variable for node s
M is the largest value at node s

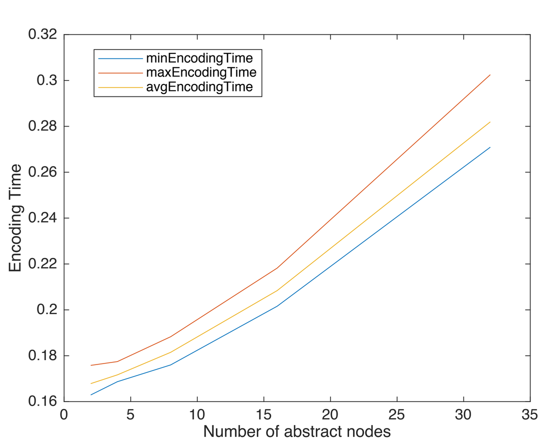
Extension of the encoding to INN

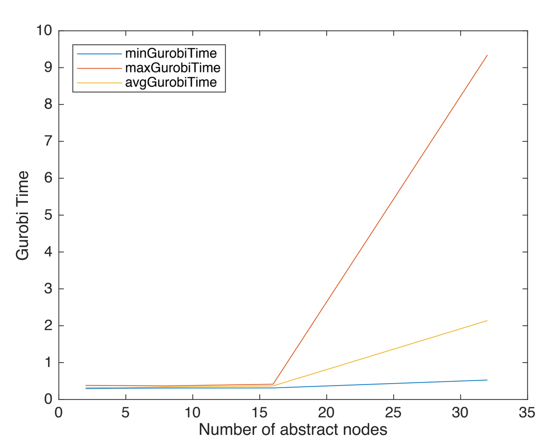
- Need to add constraints on weights and biases $W_i^l \leq W_i \leq W_i^u$, $b_i^l \leq b_i \leq b_i^u$
- Leads to non-linear constraints since W_i and b_i are now variables

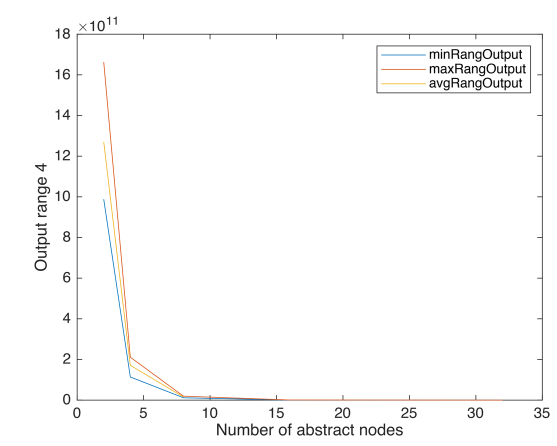
> Observation: Can safety replace W_i and b_i in the first constraint by W_i^L and b_i^L, the lower bounds on weights and biases, and in the second constraint by W_i^U and b_i^U, the upper bounds on weights and biases

Experimental Evaluation









- Evaluation on a ACAS Xu benchmark with 6 hidden layers and 50 neurons in each layer
- Abstraction, encoding and MILP solving times increase and precision decreases with the increase in the number of abstract nodes
- The times and precision have vary based on the partitioning of the nodes for a fixed number of abstract nodes

Conclusion & Future Works

- **Conclusion:**
  - Our experimental results demonstrate the usefulness of abstraction procedure to compute the output range of the neural network
  - It shows the trade-off between the precision of the output range and the computation time
  - The precision of the output range is affected by the specific choice of the partition of the concrete nodes even for a fixed number of abstract nodes
- **Future Works:**
  - Exploring different partitioning strategies for the abstraction with the aim of obtaining precise output ranges
  - Consider more complex activation functions
  - Analyzing the interval version of the neural network for these new activation functions

Acknowledgements: National Science Foundation CAREER Award and Office of Naval Research Young Investigator Award