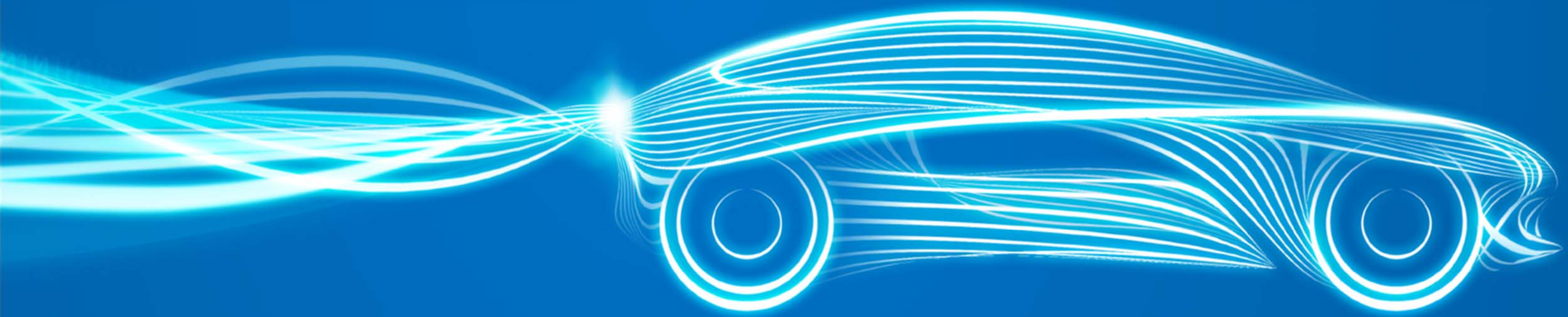
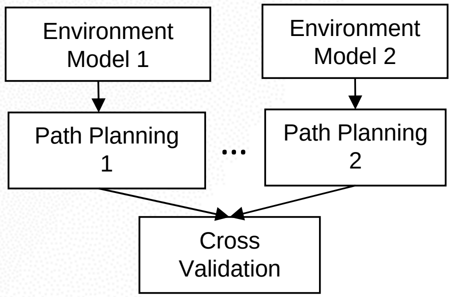


On finding low complexity heuristics for path planning in safety relevant applications

Krutsch Robert, Intel

Problem statement

- Learning based methods for path planning (e.g. based on reinforcement learning) have no strong theoretical guarantees in comparison to search based methods. Even given a perfect environment model output can be wrong.
- In current implementations multiple environment models and multiple planners are used to get to desired functional safety levels and also to have a robust system.

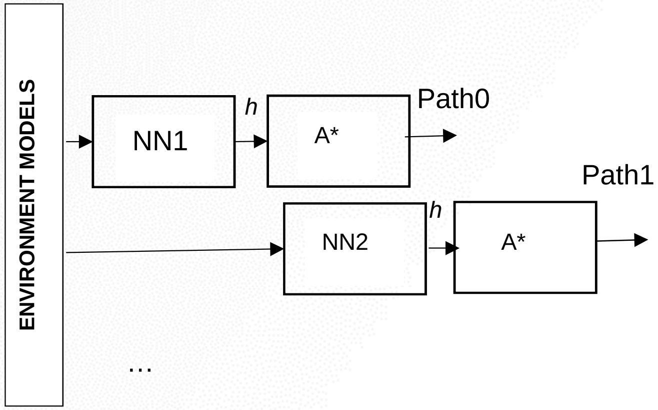


Questions addressed in the paper:

- Can we combine search based methods with learning based methods such that we can theoretically have human like behavior in desired cases but also strong theoretical guarantees ?
- Can we find a way to minimize the impact on performance in cases where we need diversity due to functional safety reasons ?

Proposal

- Train a fully convolutional neural network of similar form as used in pixel labeling applications to generate a heuristic for a search algorithm
- The input to the neural network is the Euclidian distance heuristic, the occupancy grid and the start and stop positions
- The optimization is done by balancing desired behavior against optimality; it was observed that direct optimization is prone to instabilities
- When implementing on an SoC we can pipeline NN processing on an accelerator with search on general purpose cores and hide extra compute time



$$L_1 = \frac{|h_o - h|}{|h_e - h|}$$

$$L_2 = |h_o - h| + Relu(|h_e - h|)$$

h_o - Desired behavior, true distance to the goal in the paper
h - Output of the neural network
h_e - Euclidian distance heuristic

Results

- We have generated a data set with ~5*10^5 occupancy grids with perfect heuristic to serve as input
- We compared the Euclidian metric heuristic against the heuristics from L1 and L2 by looking at the number of opened nodes, computational time and the length of the path found.
- ***C*** and ***T*** should be positive and ideally high; ***O*** should be small, ideally zero

$$C[\%] = \frac{\sum(OE - ONN)}{\sum OE} * 100$$

$$O[\%] = \frac{\sum(DNN - OD)}{\sum OD} * 100$$

$$T[\%] = \frac{\sum(T_{old} - T_{new})}{\sum T_{old}} * 100$$

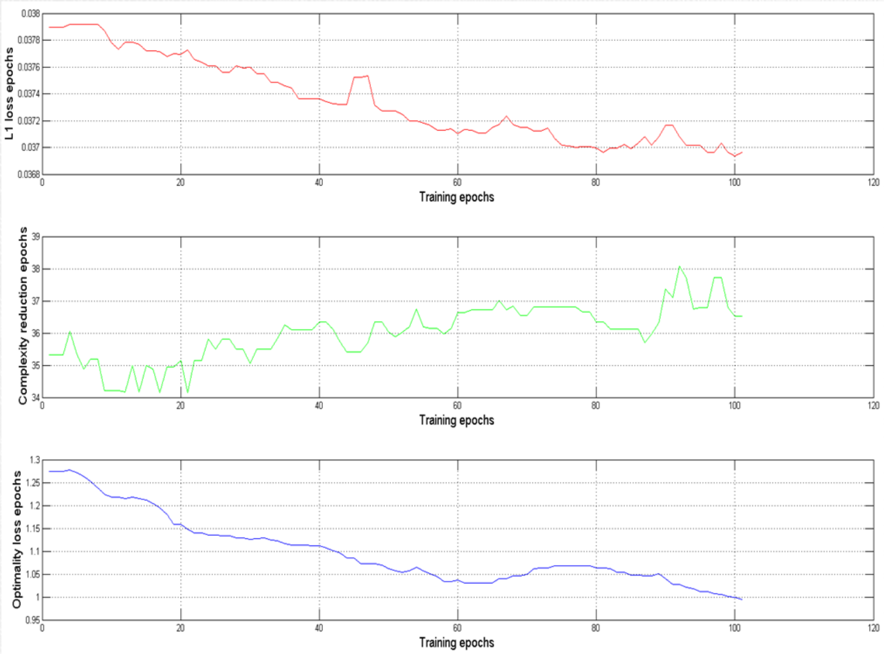
Loss Function	C[%]	O[%]	T[%]
L_1	37.64	1.01	36.32
L_2	35.82	0.71	32.45

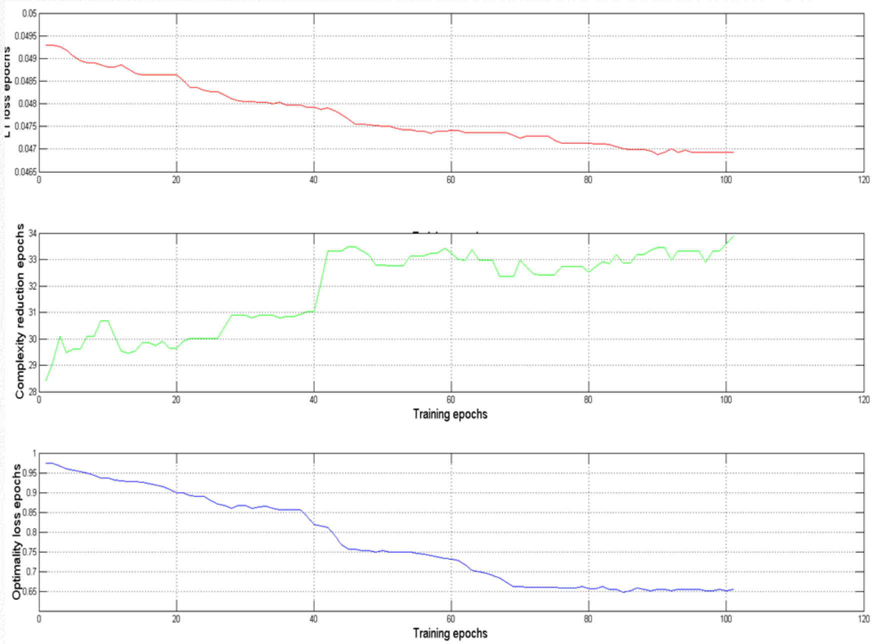
OE = Number of opened nodes by A* given Euclidian distance as heuristic
ONN = Number of opened nodes by A* given output of the neural network as heuristic
OD = optimal path length
DNN = path length found by A* given the heuristic obtained with a neural network
T_{old} – Runtime for the Euclidian distance heuristic A*
T_{new} – Runtime for the neural network heuristic A*

Results

Question:

- Are this cost functions any good ? When the cost function is minimized to we get better in terms of optimality and in terms of complexity ?





Thank You