Shabbir Hussain

Robert Platt

NEU CS-5335

7-Dec-2016

Cartesian to C-Space estimation: Using Neural Network as estimator

Background:

One basic challenge in motion planning is getting a collision in the path. A machine has to calculate all possible outcomes and plan a collision free route by calculating each point along the path plus many more failed routes. While a human on contrary can estimate a motion path just by looking at the Cartesian space. Inspired from human motion planning, I explored the possibility of directly mapping some Cartesian space images to a C-Space images by training a Neural Network to estimate free space vs blocked space categorization in nd C-Space($n = 2$). We hope to take advantage of the fact that robot configuration is fixed, so the way it (and all joints in it) interacts with an obstacle at a distance x should be similar for all similar objects at a same distance x.

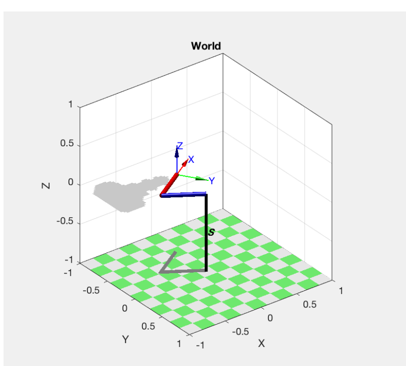


Figure1 :Environment

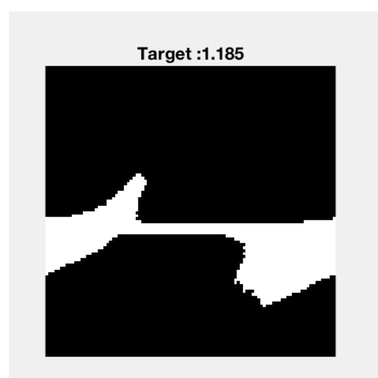


Figure 2 :C-Space

Project Setup:

For this project an environment of a planar robot is set. This robot has two joints and two links of fixed width as shown in Figure1 :Environment. Obstacles are planar which are placed randomly around our robot. Obstacles are random shapes fetched from online dataset [1]. These shapes are then placed randomly around robot in at most two quadrants. We then saved every experiment as an image with robot at the center (Robot not printed to image)

Next, we calculated corresponding label for each image by sampling each joint angle 100 times and calculating if any part of robot is in collision of the obstacle or not. Every collision is marked as one (white) and free space is marked as zero (black) as shown in Figure 2 :C-Space.

Learning:

Moving towards goal of learning projection we set up a neural network with pixels of image as input and pixels of the label image as output. Our starter model has just one layer architecture similar to a linear regression as shown in Figure 3 :Neural Network.

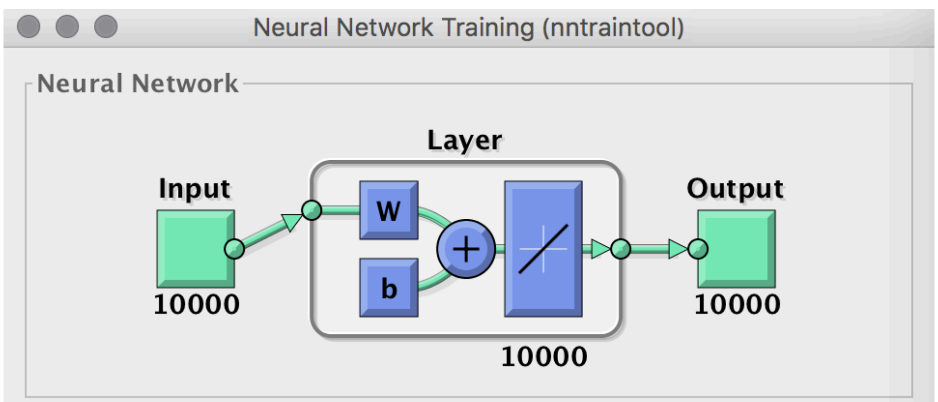


Figure 3 :Neural Network

We created different resolutions of input and output images of calculations to compare model accuracy and performance as follows.

- 10x10
- 30x30
- 50x50
- 100x100

As expected 10x10 models are faster for training and predicting with an average error rate of 0.4%. On the other hand, the 100x100 resolution models are much more accurate in predicting the C-Space. The average accuracy of 100x100 model is 99.999% as seen from the Figure 4.

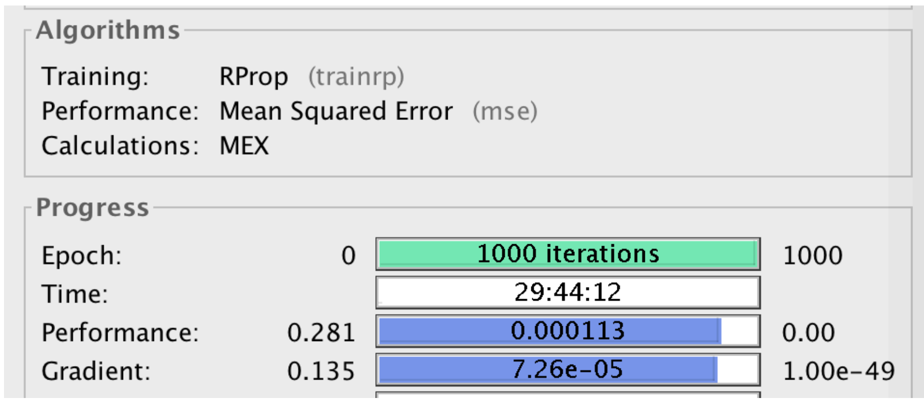
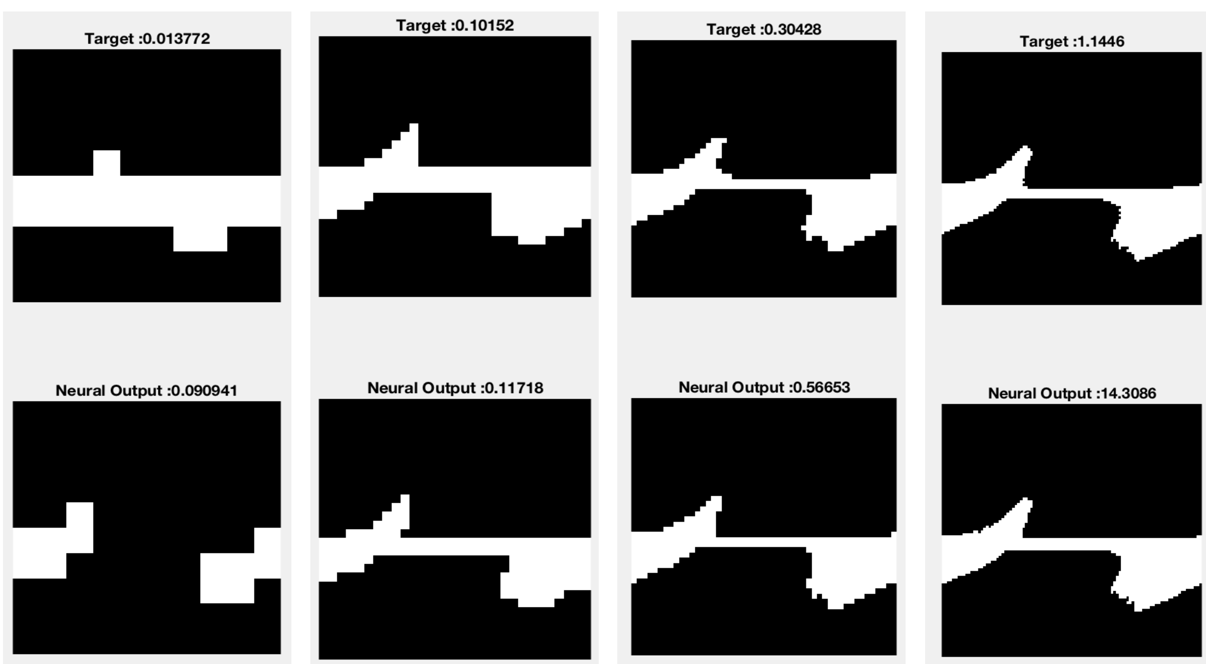


Figure 5 :Performance for 100x100

Below Figure 4 :Resolution Performance Comparison shows the comparison of performance in predicting the C-Space with models of different resolutions. We can see continuous improvement with increase in resolution of input and output image. Although neural network shows serious



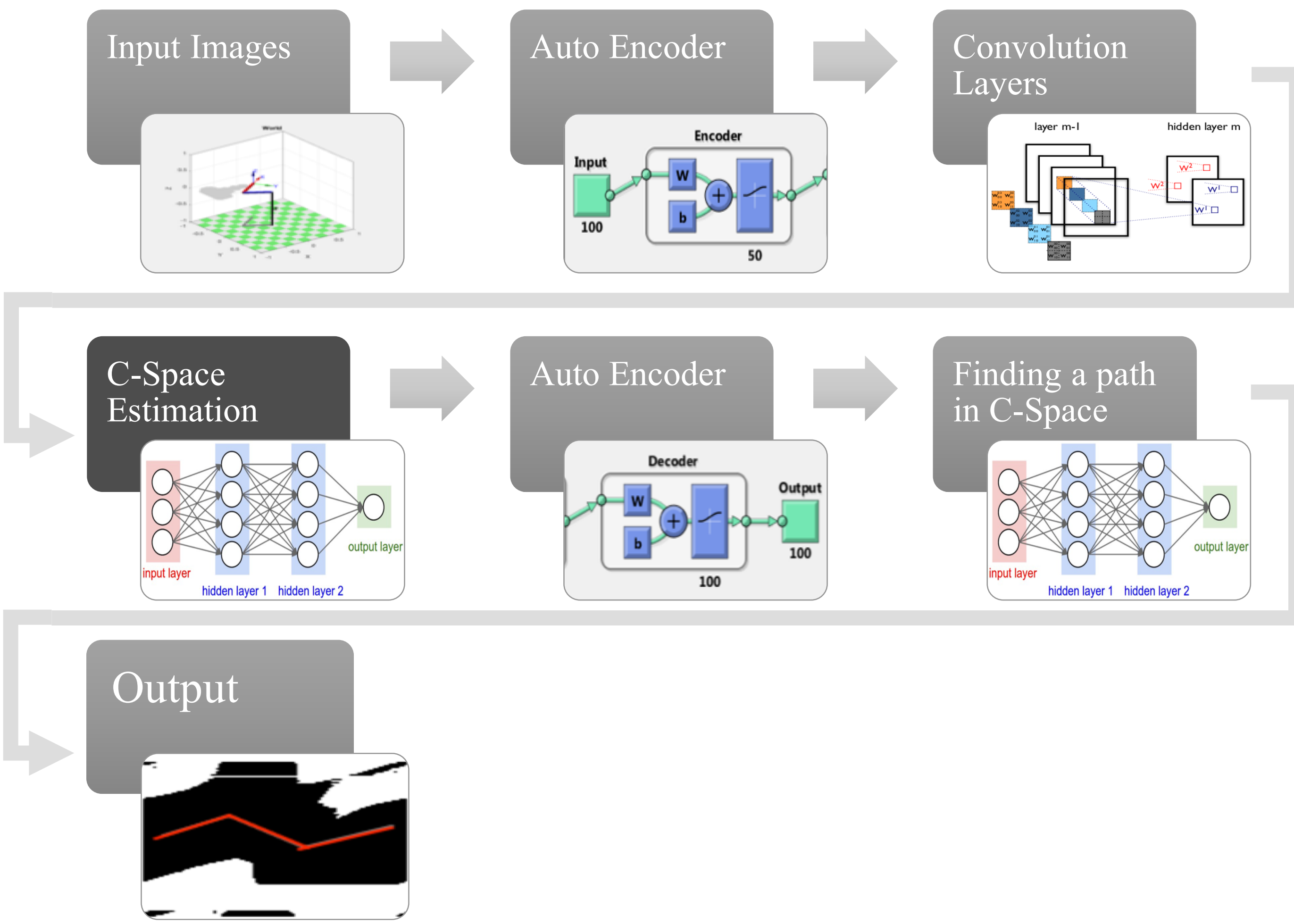
10x10 **30x30** **50x50** **100x100**

Figure 4 :Resolution Performance Comparison

performance penalties once you cross in higher resolution this could be improved using Auto Encoders to reduce the dimensionality of the inputs reliably. Numbers in the header label of “Target” and “Neural output” shows time taken for the calculation.

Future Scope:

This project can be expanded to train a full scale robot with multiple joints to predict feasible joint configurations. This could make process of path planning intuitive for every robot configuration. I would like to explore the possibility of action planning using pure neural networks trained on various aspects. A proposed design below could yield a performance balanced output using Auto Encoders.



Works Cited

[1] "Shapes Dataset," [Online]. Available:
http://www.dabi.temple.edu/~shape/MPEG7/dataset.html.