Neural networks in path planning

Anshasi Said
Departement of Computer sciene
Technical university of deggendorf
Deggendorf, Germany

Abstract
This paper presents the use of neural networks in path planning, a U-net neural network is used to achieve this goal, the neural network receives its environment obstacles, as well as the starting, final destination and optimal path from a database, this work is demonstrated using a supervised learning approach, a part of this paper will demonstrate path planning for an initial point residing the configuration space to a destination point, the other part will describe path planning for a 4 DOF robotic arm. The implemented neural network was able to solve the path planning problem for numerous sets of environments efficiently.

I. Introduction

There are various methods that have been used for path planning, the classical methods of sampling-based path planners such as RRT and PRT have been used, however if the configuration space was big and high dimensional, sampling-based planners suffers from high computation times to plan a path, also optimization path planners have been introduced such as OMP which optimizes the trajectory by solving many inverse kinematics problems simultaneously and sets the trajectory within the convex hull, this guarantees that the optimized trajectory doesn't violate the joint constraints however the convex hull property of OMP doesn't guarantee that the trajectory is collision free with the geometry. The purpose behind using neural network for path planning is to introduce a method which reduces the time needed to find the shortest path between the initial and final destination with regard to other approaches.

II. Methods

A. Neural network architecture

Our neural network architecture is a combination of fully convolutional neural network and a fully connected neural network, the fully convolutional neural network comprises of an encoder part and a decoder part, the fully connected part contains one dense layer. For path planning of the robotic arm, the auto encoder receives the obstacles, the starting position and final destination of the 4 DOF robotic arm as input features and the optimal path of the robotic arm as input labels, the output of the convolutional network is then feed to a fully connected dense layer for mapping the output layer of the convolutional network to a dimension that matches the label, in the case of Robotic arm a dimension of (4X20) is required. The same applies for point path planning, however the output label dimension is converted to (2X22). We kept the number of dense layers at minimum, as adding extra dense layers increased the number of trained parameters significantly, which required big size of training data set and would increase the time needed for training the model.

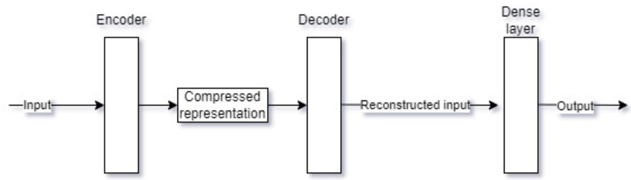


Fig. 1: Neural architecture consists of an autoencoder and fully connected layer.

B. Data set description

Two data sets have been used for producing results, data set for point-to-point path planning contains 5000 worlds, i.e., 5000 different environments, each of these environments has 1000 different initial and final destinations with an optimal path connecting them. On the other hand, the data set used for 4 DOF robotic arm path planning consists of 10000 worlds, the number of initial/ final destination including the path connecting them varies for each environment.

C. Neural network training

For point-to-point path planning the neural network has been trained on 10 worlds representing 10000 different images, for 100 epochs. For 4 DOF robotic arm the network has been trained on 3 worlds for 100 epochs representing 3 different environments, each environment contains 325 different starting and end states and a unique path connecting each of these states.

III. Results and Discussion

Both use cases demonstrated in this paper is bound to 2-dimensional space, i.e., motion of the point as well as the robotic arm is bound to 2-dimensional space, the point path consists of 22 coordinates points including the start point and end point. The path of the 4 DOF robotic arm, consists of 4 coordinates points which make up the 4 DOF robotic arm at starting position, 4 coordinates at final position and 18 set of 4 points resembling the path in between.

A. Point to point path planning

Using U-nets allowed us to extract more features while keeping the number of parameters small. The neural network had 2,121,373 million parameters that need to be trained, after training the network on 10000 images, we achieved a training loss of 0.0819 and a testing loss of 0.15. Fig-2 shows the plotted training loss.

The model developed for point-to-point path planning gave us pretty good results, and the planner was able to plan shortest path between starting and goal point. Figure-3 shows the predicted path of our model. However, our model didn't take into account the tolerance between the obstacles and the planned path, as a consequence, the path planned was very close to the obstacles in some samples.

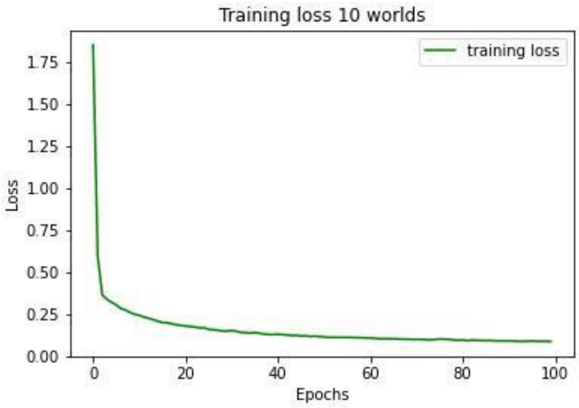
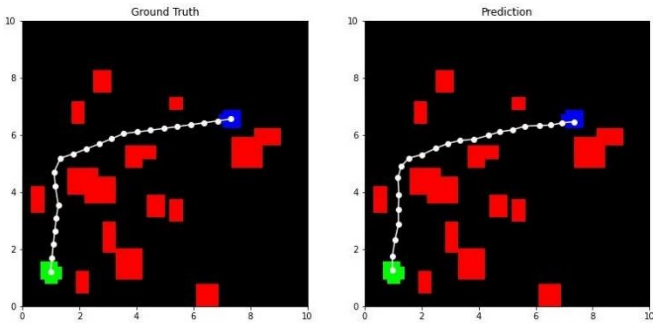
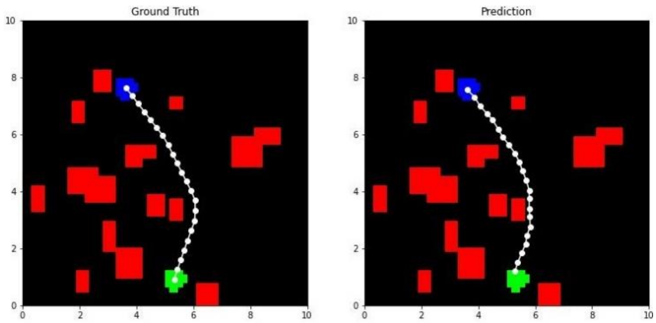


Fig. 2 Training loss for point-to-point path planning amounts to 0.0819.



(a)



(b)

Fig. 3 (a, b): Ground truth vs prediction for point-to-point path planning for two different environments.

B. 4 DOF robotic arm path planning

The neural network has 2,268,856 million trainable parameters. After training the network on 3 different worlds, a training loss of 0.1975 and a testing loss of 0.3198 has been achieved. Figure-4 shows the plotted training loss for the 4 DOF robotic arm model.

Although the same network architecture has been used for path planning both point-to-point and 4 DOF robotic arm, the training loss attained in the 4 DOF robotic arm isn't as good as that in the point-to-point case, and this was due to different reasons, the first reason is that, the network was trained on lower number of worlds. This was done because the number of robotic arm paths that belongs to a single world varies from one world to another, and this made it difficult to filter which path belong to which world since 10000 worlds were available in the data set, which needed a lot of data preprocessing beforehand.

The second reason why the predictive model in the robotic arm use case wasn't as good as predicting point-to-point path, is the fact that there exist samples within one world, where robotic arm would collide with the edges of the environment geometry, also in many worlds there was an overlap between the robotic arm and the environments obstacles. This has restricted us to train the model only on worlds where there exist minimal robotic arm/obstacles overlap.

Even with above constraints, the model was able to predict an acceptable path for some samples. Figure-5 shows path prediction for two different worlds.

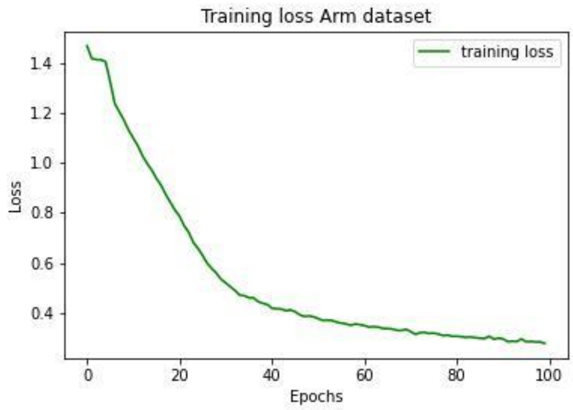
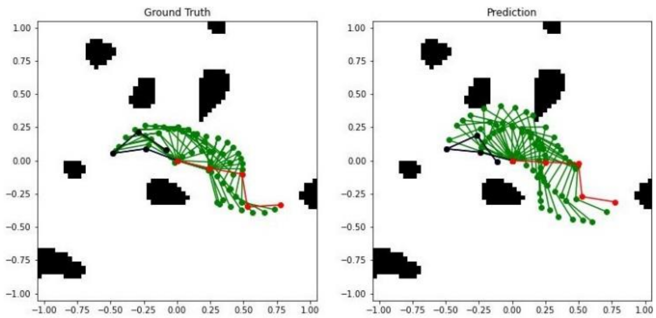
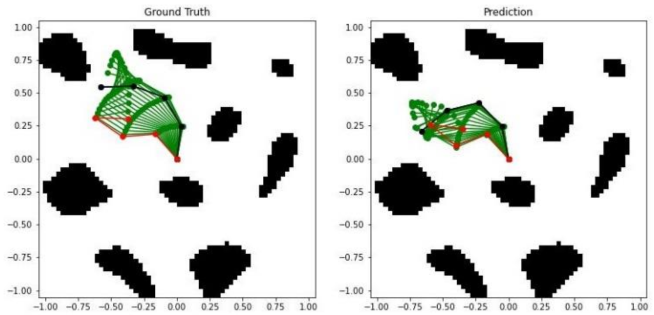


Fig-4: Training loss for 4 DOF robotic arm path planning amounts to 0.1975.



(a)



(b)

Fig. 5 (a, b): Prediction of the neural network for the 4 DOF robotic arm for two different environments.

IV. Conclusion

The model architecture developed has proved itself to be reliable for path planning problems, had it been trained on high quality datasets, however there are still aspects where it can be improved. For instance, the model didn't take into account the volume the robotic arm would occupy in real environments. Also, the use case can be extended to 3-dimensional spaces.

V. REFRENCES

[1] Ahmed H. Qureshi, Yinglong Miao, Anthony Simeonov and Michael C. Yip "Moion planning Networks: Bridging Gap between learning-based and classical motion planners".

[2] Kevin M. Lynch and Frank C. Park "Modern Robotics: Mechanics, Planning and control".

[3] Russ Tedrake "Robot Manipulation: Perception, Planning and Control".

[4] Matt Zucker, Nathan Ratliff, Anca D. Dragan, Mihail Pivtoraiko, Mathew Klingensmith, Christopher M. Dellin, J. Andrew Bagnell and Siddhartha S. Srinivasa "CHOMP: Covariant Hamiltonian Optimization for Motion Planning".