

Learning Formulation Optimisation for Obstacle Avoidance in Autonomous Racing

Benjamin Evans

1

2 Introduction

This paper explores the effect of training parameters on an agent learning the task of unmapped obstacle avoidance in autonomous racing. For the study, we use the serial agent planner presented in [1] as our navigation stack.

3 Evaluation Methodology

The planner is trained and tested in simulation using the official open-source F1/10th simulator ¹.

3.1 Metrics

The two main metrics are performance and success rate, since these are the measure of racing performance. The sample efficiency, repeatability and robustness of the policy are also measured.

We measure the performance of the agent after training it in batches of 10,000's. When the performance difference between subsequent batches is negligible ($< 1\%$) then we assume the network has converged.

3.1.1 Robustness to Sensor Noise

In the real world, all sensor measurements are noisy. Therefore, we evaluate the ability of our agents to perform in noisy environments. 5, 10 and 20% noise values are added to the measurements and the results measured.

3.2 Learning Variables Studied

3.2.1 Neural Network Architecture

The neural network architecture is studied along the two parameters of the number of neurons and depth of layers. A preliminary study on the kind of activation function is also conducted.

3.2.2 Reward Signal

The reward signal is used to encode the desired racing behaviour in an equation that tells the agent how good or bad it's performance is. The follow five reward signals are considered:

1. Progress-based reward
2. Velocity-based reward
3. Action-based (minimum steering) reward

¹Available online: www.GitHub.com/

4. Distance to obstacle
5. Distance to boundary

3.2.3 Reward Hyper-parameters

The progress-based reward signal is then selected for further study. The reward consists of three different terms that are individually adjusted to determine their effect. An additional change is to consider the distance squared at each timestep. $\gamma (1+i)^2$

3.2.4 Algorithm

The algorithm used to train the agent is evaluated. The main comparison is between the two state of the art algorithms for continuous control, SAC and TD3.

3.2.5 Considerations

- Noise in training
- Replay memory size
- Learning rate?
- Batch size?
-

3.3 Physical Parameters Studied

3.3.1 Number of Range Finders

We experiment with how the number of range finders affects the training performance. 5, 10, 20 and 40 range finders are used. With 10 range finders, we vary the field of view from $\pi/2$ to π . We also consider the effect of the max range finder range, 3, 5 and 10 meters are used.

Range finders in real life include a certain amount of noise. We consider adding different quantities of noise in the training process and then measure the resulting policies to robustness to noise in readings.

3.3.2 Stacking Windows

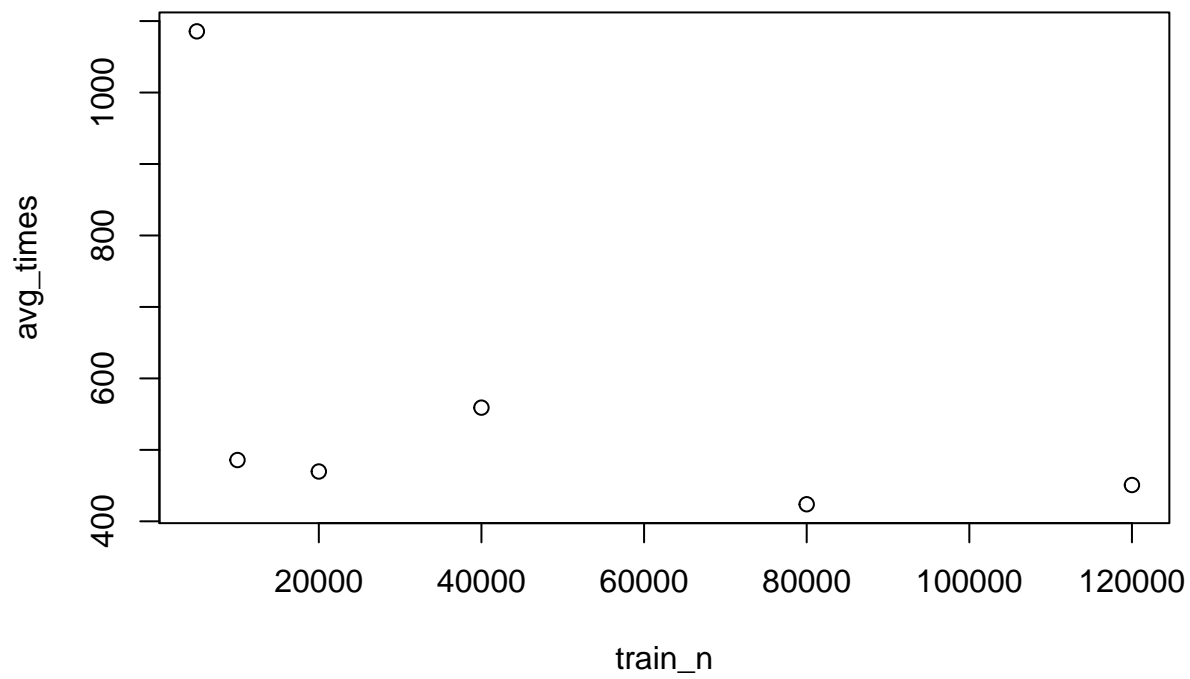
We evaluate the effect of stacking multiple windows on top of each other. 1, 2, 5 and 10 windows are used in the analysis.

3.3.3 State Variable

The original state variable consists of 14 values. We remove each value and consider what effect it has on the training.

4 Results

```
test_data = read.csv("DataTable.csv")
attach(test_data)
# plot(train_n, success_rate)
plot(train_n, avg_times)
```



5 Conclusions

6 Bibliography