

Learning Formulation Optimisation for Obstacle Avoidance in Autonomous Racing

Benjamin Evans

1 Introduction

This paper explores the effect of training parameters on an agent learning the task of unmapped obstacle avoidance in autonomous racing. The serial agent planning architecture is used for the evaluation. For each study, tests are run for a set of perturbations from baseline values. Several parameters are chosen for study.

1.1 Evaluation Methodology

The simulations are run in the custom built F1/10th simulator. The task is to use a LiDAR scan and the vehicles current heading and velocity to determine the steering angle that will lead the vehicle to the end of the forest without colliding with any obstacles.

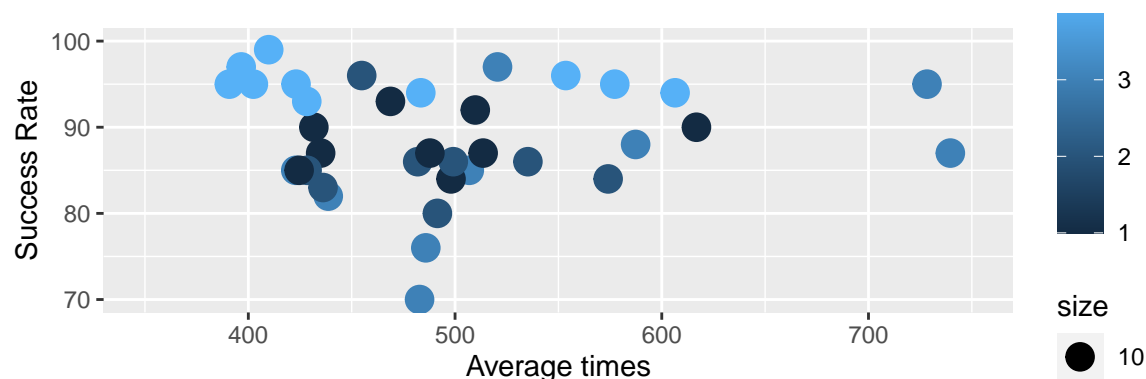
The base learning formulation uses a reward signal that rewards progress along the track. The track is a straight 20m long section of road with 4 obstacles randomly spawned on each episode.

1.2 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.

2 Repeatability & Distribution

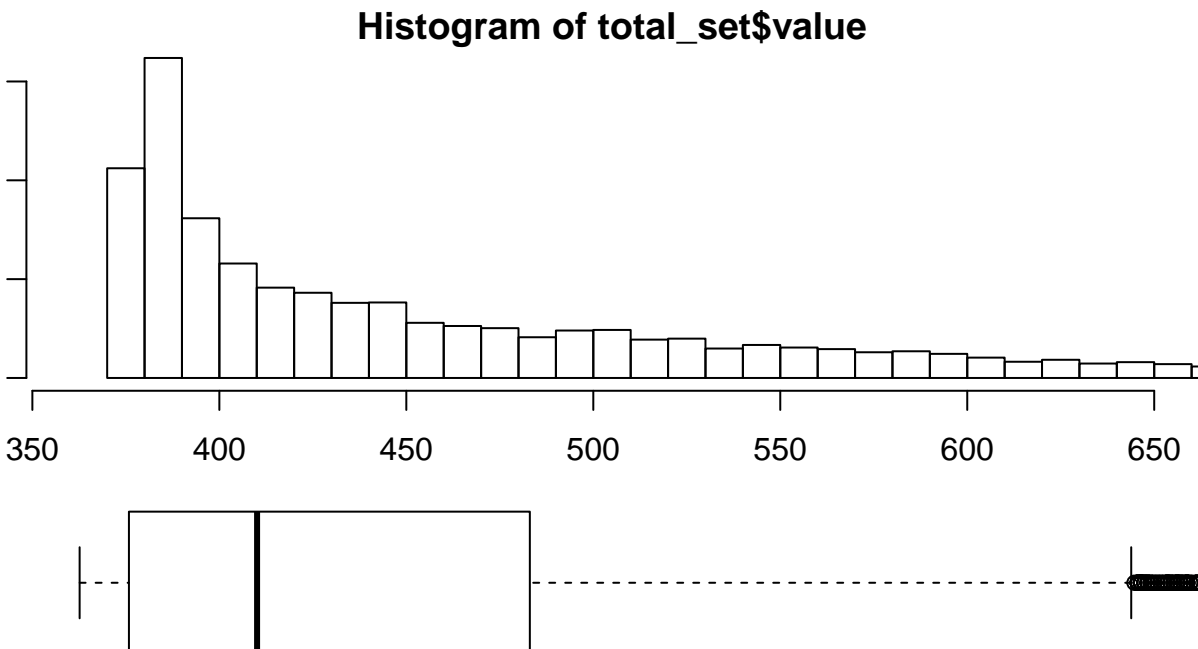
The repeatability of the training and testing is studied to measure the statistical significance of the results. An agent is trained 10 times with the same parameters and the distribution of the results is studied.



The figure above shows the distribution of the average times and success rates from agents that are trained using the same parameters. Note that the different colors represent different tests with slightly different parameters.

The distribution of the resulting lap times is measured by testing the 10 agents trained for the repeatability and running 1000 test episodes of each of them. The combined distribution is shown below.

No id variables; using all as measure variables



The figure shows that a significant majority of the results are around the lower lap times (below 400ms). The x axis represents the time to complete an episode in ms. The data is skewed to the lower lap times with the majority of times being within 380 to 390ms.

3 Learning Parametes

The learning setup parameters of the number of beams used in the LiDAR scan, the number of training steps trained for, and the size of the hidden layers of the network are evaluated for their effect on the results.

3.1 Neural Network Architecture

The number of neurons in the two hidden layers of the networks is changed to determine if this has an effect on the agents ability to learn an accurate policy. A similar approach, using 60 range finders and several state variables, used networks with two hidden layers of 256 neurons. Considering that we are generally using few range finder readings, we perturb this value downwards to see if it has any effect on the outcome.

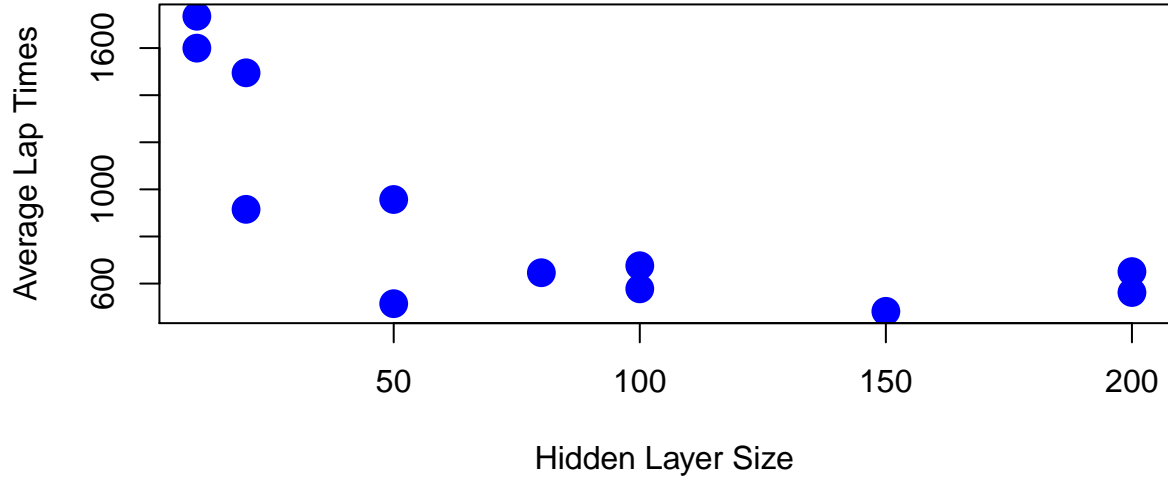
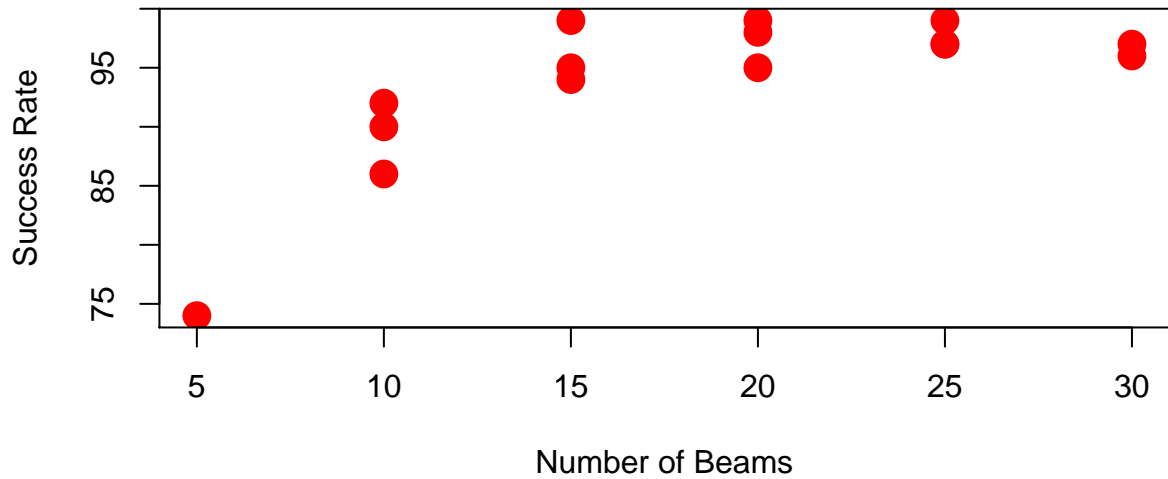


Figure @ref(fig:h_size_fig) shows how for different hidden layer sizes after 100 neurons, there is a negligible change in the performance of the vehicle. For smaller network sizes, the average lap times become a lot worse. Therefore, 100 neurons is chosen as the size to be used.

3.2 Number of LiDAR Beams

LiDAR sensors provides dense spatial data of the surrounding environment and can have up to 1000 beams. Only several beams are sampled from the full scan to keep training times fast and the network small. Originally 10 beams were used to build on work by Tai et al.

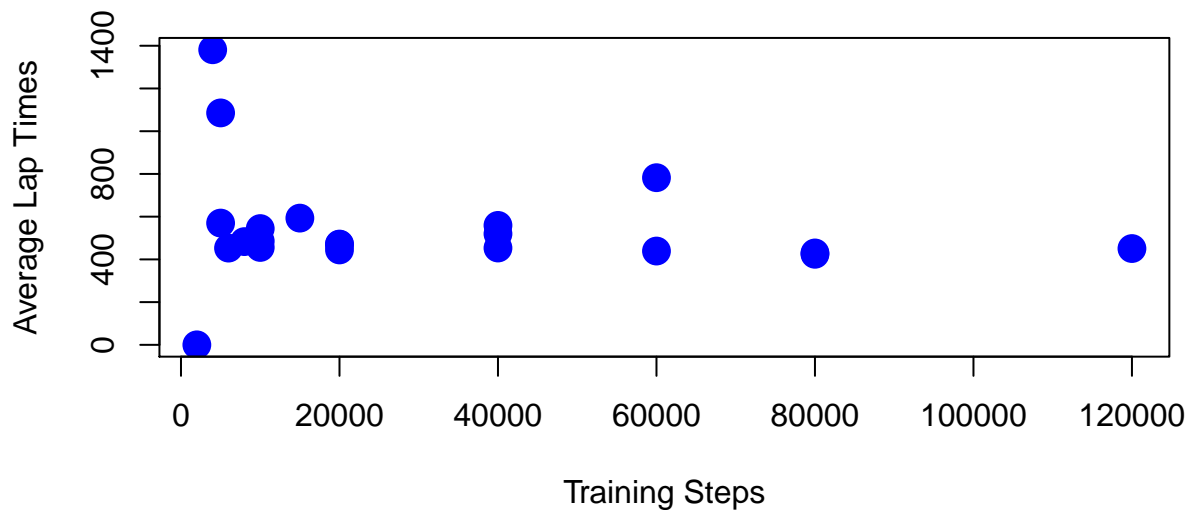


The results show that using 15 range finders has a marked difference over using only 10. Above 15 range finders the performance levels off. Therefore, 15 range finders are selected for use.

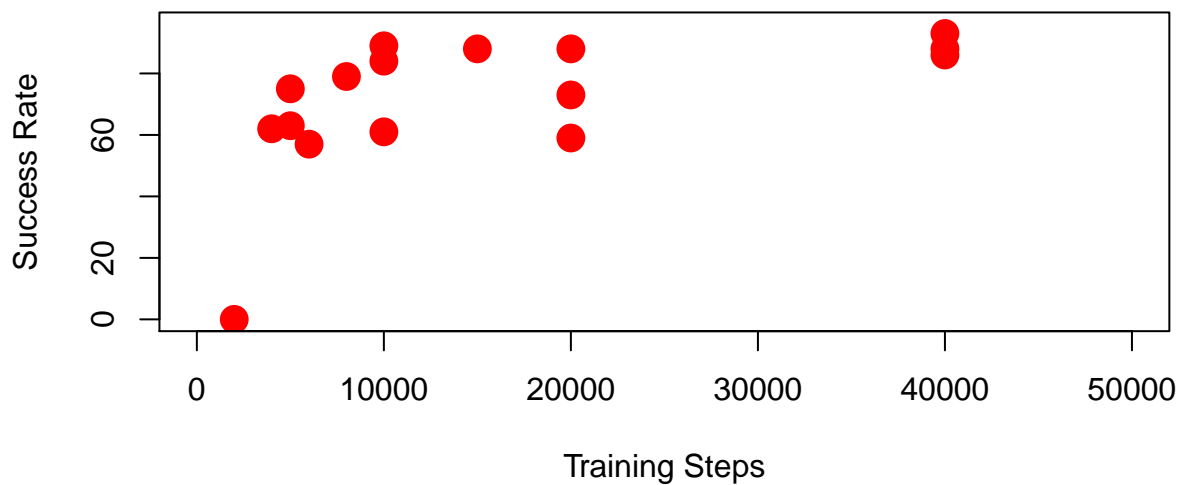
Further study on the LiDAR scan could look at the distance used for scaling and clipping, the effect of noise in the readings and the field of view.

3.3 Training Steps

While reinforcement has the advantage of being able to learn from the agent's own experience, it is often sample inefficient. The number of steps required to train the agent to convergence is measured.



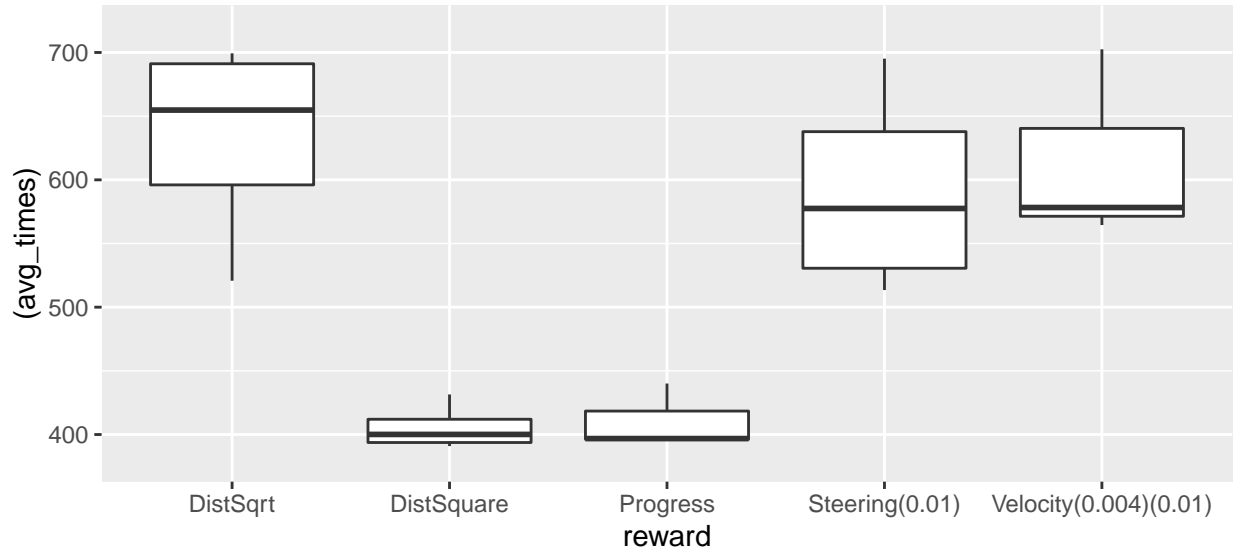
The graph shows that the performance levels out after 20,000 training steps. Therefore, the training time is decreased. The next figure provides a more zoomed in look at the success rate for shorter training steps.



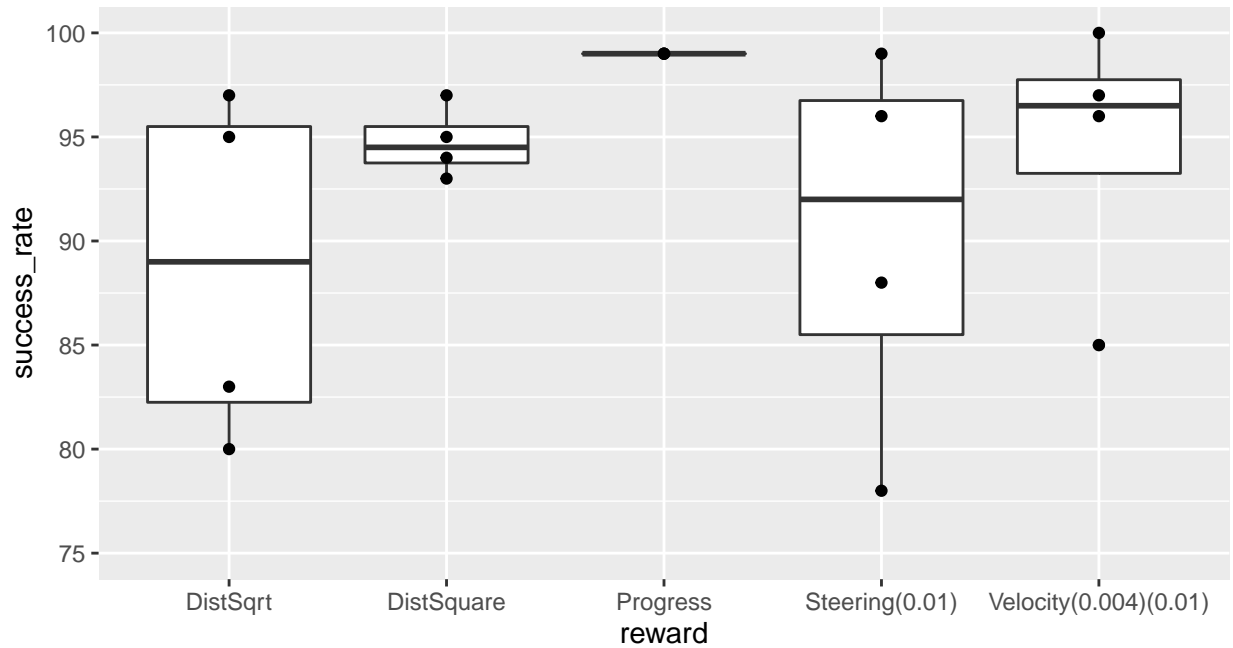
4 Reward Signals

4.1 Comparison

Five different reward signals are selected and used to train agents. The table below shows the average results from training 5 agents with each reward signal.



The figure above shows how the progress and DistSquare reward signals produce the fastest lap times by over 100ms. Therefore, they are selected for further comparison below.



The figure above shows how the success rate of the progress reward is consistently much higher than the other reward signals. This is possibly due to hyper-parameter balancing.

4.2 Progress 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.

5 Future Studies

The follow future studies can be carried out: - Algorithm used, specifically TD3 vs SAC - Window stacking, does stacking 2 or 3 windows help? - Effect of each state variable, are the current velocity and steering needed?

6 Conclusions

This report focused on the evaluation and tuning on the serial planning architecture for unmapped obstacle avoidance. The table below presents the final values that were selected to produce the best performance.

| Parameter | Value |
|-------------------|----------|
| Hidden layer size | 100 |
| Training steps | 20000 |
| LiDAR beams | 15 |
| Reward Signal | Progress |