Reinforcement learning for simulated homing using computer vision

Reinforcement learning is a class of machine learning algorithms in which we have an agent that’s primary goal is to interact with an environment and seek to maximise the cumulative reward from taking actions within that environment. The agent is not implicitly told what actions will lead to positive rewards however it must learn these through experience (Richard S. Sutton 2020 1.1). This is both an advantage and a disadvantage of reinforcement learning as it allows the agent to find novel solutions to problems which we may have not considered however the time to find such solutions can be and often is time consuming.

Reinforcement learning has also been combined with deep learning techniques so that the agent can learn from high dimensional input such as video and images. (Volodymyr Mnih 2013) is one of the most renowned examples of this where an agent was trained using a deep Q network to learn state of the art policies to play six out of seven Atari 2600 games using the game pixels as input. The RGB pixels were used as input to a convolutional neural network which allowed the algorithm to optimise the filters for feature selection rather than the filters having to be engineered using prior knowledge. This independence allows the algorithm to hone in on patterns in the input that humans may not be able to discover or perceive. The deep q network used here would be classed as end to end as the input and reward system are interlinked.

Others have successfully made models that can achieve the task of homing and autonomous navigation however most if not all these models rely on some form of mapping to allow the robot to navigate the environment. Examples of this would be self-driving vacuum cleaners such as Roombas. These devices use a plethora of sensors to be able to detect objects and cliffs as well as estimate the robot’s current position in a space and use this to create a virtual map of the environment. This technique is known as SLAM (simultaneous localisation and mapping). This approach while appropriate in most environments might not be suitable for dynamic environments in which obstacles may move seemingly random to the device. An example of such an environment may be shop floors or restaurants that may change seating areas depending on reservations as group sizes may vary.

The work proposed by Pengpeng Zhang (2019) shows an autonomous robot equipped with deep reinforcement learning techniques that allow for maples navigation however this approach uses 10-dimensional laser readings as their sensory input rather than visual data.

This method while allowing for autonomous navigation doesn’t teach the robot about how to navigate an environment using visual imagery. A combination of visual imagery and laser sensor data could be used to create a model that would port over to a real world environment more effectively as images alone may not generalise well to object avoidance however laser sensor data would.

Liulong Ma, Yanjie Liu and Jiao Chen (2019) propose a model for maples robot navigation using RGB images as input. Their approach used 3 convolution layers to extract features from the input as well as two fully connected dense layers to determine Q values for the discrete action space. This was further improved by decoupling the visual feature model from the Deep reinforcement learning network which allows for easier migration to new environments according to the authors.

The model learned quickly and efficiently in simulated environments however this is not a strong indication of how well the agent will perform in the real world. The decoupling of the visual network and the reinforcement learning algorithm improved the sample efficiency and generalisation of the agent to new environments. This methodology of separating representation learning, and policy learning is relatively new and has been explored by (Adam Stooke 2021) where they show an unsupervised learning model which trains a convolutional encoder to associate pairs of observations within a short time span. The testing shows that the use of the encoder matches or outperforms end to end reinforcement learning models in most environments however the flexibility and improvement gained from reward free representational learning will likely add more complexity to the solution than it is worth especially for real world environments.

Using a simulation to train the model allows for faster training as the environments can be reset and changed quicker which in turn allows us to generate large amounts of training data for the agent to learn from with ease. We can also simulate environments that would be hard to test in the real world such as space or other planets. However, by using a simulation this also means that we have a potential for overfitting to the simulation and creating a model that would not perform well in the real world. Simulating complex environments also increases the computational cost required to train the model which may hinder the performance of the agent as the agent may not be able to process the state changes fast enough due to the extra computation required to simulate the environment.

The efficacy of using a simulation to train a reinforcement learning agent that will perform in the real world has been explored by Blazej Osinki(2020) where they propose the idea that good performance in a simulation doesn’t equate to proper function in the real world. They show that discrete action spaces generally lead to weak performance in the real world and that regularisation and model-based approaches had the best operation. The models used by Blazej Osinki(2020) were for real world autonomous driving which while relevant are subject to a lot of external factors that would not be applicable to the scope of my problem. For example, a simulation of city roads could not accurately factor in all aspects of the real world however a simulation modelling a home environment or restaurant is a lot more feasible as there are less factors at play.

References:

Richard S. Sutton, Andrew G. Barto (2020). “Reinforcement Learning an Introduction second edition”

Volodymyr Mnih (2013). “Playing Atari with Deep Reinforcement Learning”

Linhai Xie (2019). Reinforcement Learning Based Mapless Robot Navigation

Liulong Ma, Yanjie Liu\* and Jiao Chen (2019). “Using RGB Image as Visual Input for Mapless Robot Navigation”

Adam Stooke (2021). “Decoupling Representation Learning from Reinforcement Learning”

Pengpeng Zhang∗, Changyun Wei∗, Boliang Cai∗ ,Yongping Ouyang∗(2019) “Mapless Navigation for Autonomous Robots: A Deep Reinforcement Learning Approach”

Bła˙zej Osi´nski1,3 , Adam Jakubowski1 , Paweł Zie˛cina1 , Piotr Miło´s1,5 , Christopher Galias1,4 , Silviu Homoceanu2 and Henryk Michalewski3 (2020) “Simulation-Based Reinforcement Learning for Real-World Autonomous Driving”