Implementing Q-Learning for Path Finding

Ben Hynes

May 13, 2020

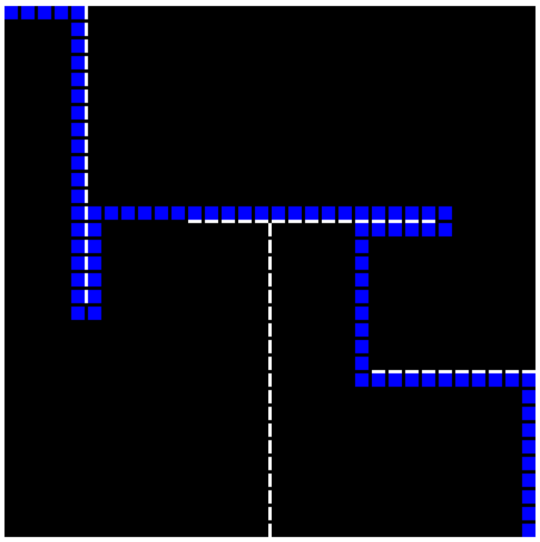

Figure 1: Final Q-Learn route.

1 Introduction

1.1 Markov Decision Process

A Markov chain is a stochastic model meaning that it is a process that involves randomness. It is uncertain which state will come after the current state. The chain describes a sequence of possible events in which the probability of each event depends only on the state attained in the previous event.

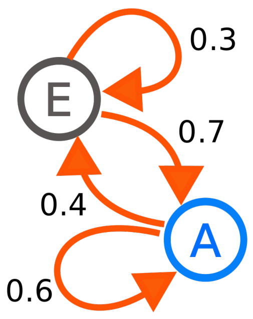

Figure 2: A two-state Markov process.

A Markov decision process is a discrete time stochastic control process. Markov decision processes provide a mathematical framework for modelling decision making in situations where outcomes are partly random and partly under the control of a decision maker. A stochastic process has the Markov property if the conditional probability distribution of future states of the process depends only upon the present state, not on the sequence of events that preceded it. A process with this property is called a Markov process.

At each time step, the process is in some state s and the decision maker may choose any action a that is available in state s. The process responds at the next time step by randomly moving into a new state s' and giving the decision maker's a corresponding reward $R_a(s, s')$.

The probability that the process moves into its new state s' is influenced by the chosen action. It is given by the state transition function $P_a(s, s')$. The next state s' depends on the current state s and the decision maker's action a. But given s and a, it is conditionally independent of all previous states and actions and therefore satisfies the Markov property.

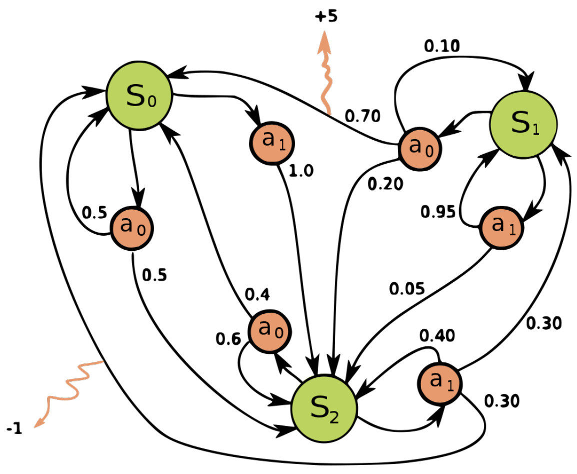

Figure 3: A Markov Decision Process with rewards in orange.

1.2 Q-Learning

Q-learning is a model free reinforcement algorithm. Model free means that it does not have access to a transition probability distribution associated with the MDP as this is what we aim to find. In q-learning, there is one q value for each possible action you could take from each state.

$$Q^{new}(s_t, a_t) = Q(s_t, a_t) + \alpha(r_t + \gamma \, max \, Q(s_{t+1}, a) - Q(s_t, a_t))$$

where

- $Q^{new}(s_t, a_t)$ is the new Q value
- $Q(s_t, a_t)$ is the old Q value
- α is the learning rate
- r_t is the reward function
- γ is the discount factor
- $max \, Q(s_{t+1}, a)$ is the estimation of future value

2 Implementation

This implementation is fully scalable, having been tested up to 32x32 pixel image sizes. In a 32x32 grid there are 1,024 states and 1,024 actions, totalling 1,048,576 state-action pairs. A reward of 1 will be given to the agent if a state is directly reachable from the current state. The destination state will have a reward of 999. All other state-action pairs have a reward of 0. The algorithm for training the agent and calculating the Q table can be seen below.

Algorithm 1: Updating Q values

```
for $N$ do
    set $s$ to random state;
    for each $a$ from $s$ do
        if $r_{s,a} > 0$ then
            add to list of playable actions;
        end
        set $s_{t+1}$ to state from random action choice;
        calculate temporal difference;
        update q value for $s_t, a_t$;
    end
end
```

3 Results

Finding a balance between exploration and exploitation can be difficult. Several tests were run for various values of N, the number of training moves the agent makes. The graphs on the right show the rate of learning (specifically, the average temporal difference) as these training moves are made. This decreases over time as the agent learn about the environment.

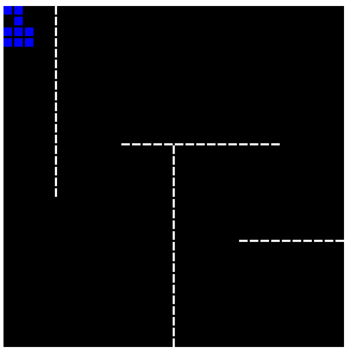

Figure 4: N = 100K

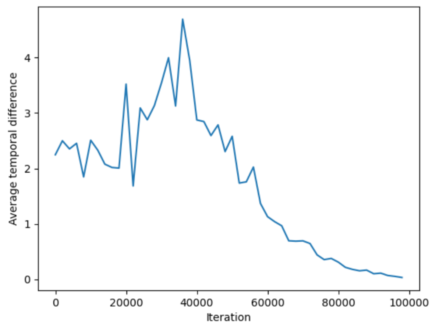

Figure 5: Average temporal difference

Increasing N to 100k allows the agent to find an optimum path.

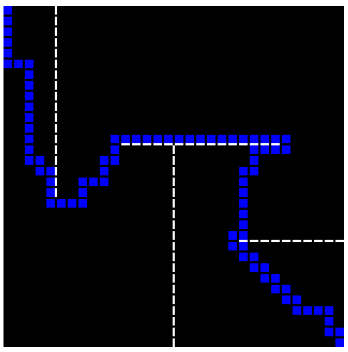

Figure 6: N = 250K

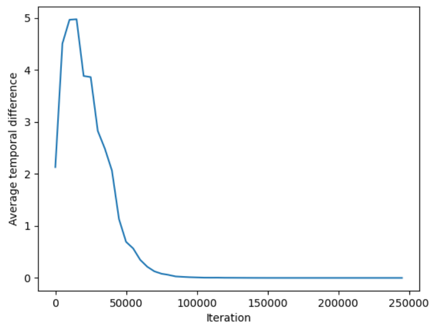

Figure 7: Average temporal difference

Interestingly, although the agent has found the shortest route, further training it will cause it do suggest a route with the minimum number of turns.

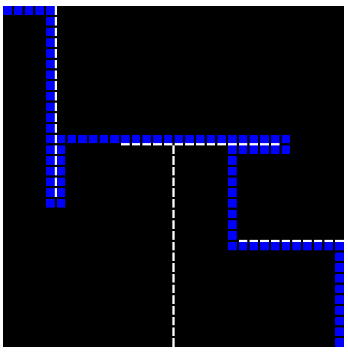

Figure 8: N = 1M

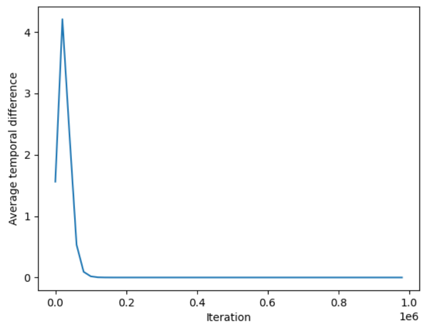

Figure 9: Average temporal difference