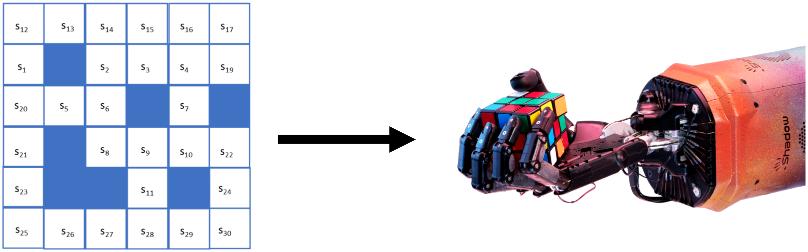
Part 2: Motivation for Function Approximation

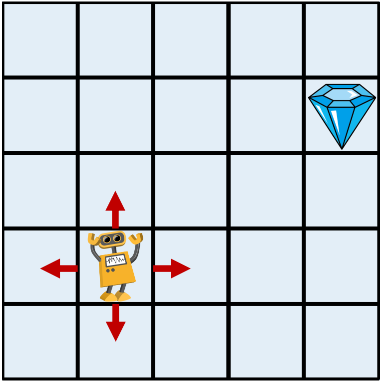
Tabular Solutions vs Approximate Solutions

- Tabular solutions: Everything you have learned so far
- Approximate solutions: Everything you will learn from now
- The theory from the first half will now be applied to real-world applications

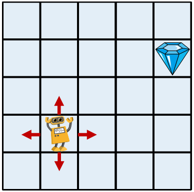


Recap of Tabular Q-Learning

- Discrete state space
- 25 states
- Discrete action space
- 4 actions
- If S is the current state, and G is the goal state, reward $R = -||G - S||$



Recap of Tabular Q-Learning



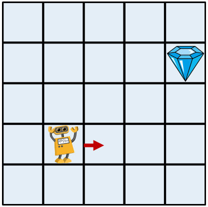
for each episode do
  $S \leftarrow S_{init}$
  for each step of episode do
    Choose A from S using policy derived from Q (e.g. ϵ-greedy)
    Take A, observe R and S'
    $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma \max_a Q(S',a) - Q(S,A)]$
    $S \leftarrow S'$
  end
end

Recap of Tabular Q-Learning

	Move Up	Move Down	Move Left	Move Right
State 1				
State 2				
State 3				
State 4				
State 5				
State 6				
State 7				
State 8				
State 9				
State 10				
State 11				
State 12				
State 13				
State 14				
State 15				
State 16				
State 17				
State 18				
State 19				
State 20				
State 21				
State 22				
State 23				
State 24				
State 25				

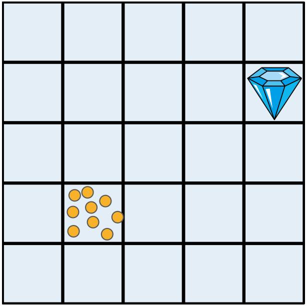
Recap of Tabular Q-Learning

	Move Up	Move Down	Move Left	Move Right
State 1				
State 2				
State 3				
⋮				
State 17				Q(S=17, A=R)
⋮				
State 23				
State 24				
State 25				



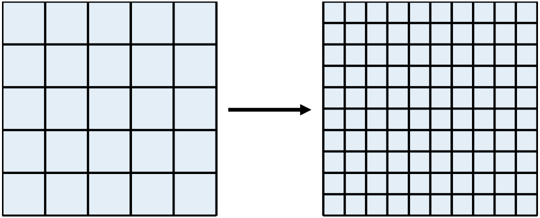
- Each entry in the table represents one Q-value, $Q(S, A)$
- But is this a good approximation of the real world ... ?

Recap of Tabular Q-Learning



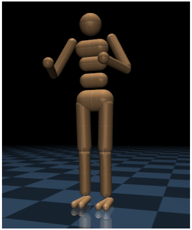
- In continuous space, one $Q(S, A)$ is the same for multiple states
- This limits our ability to accurately represent the real world

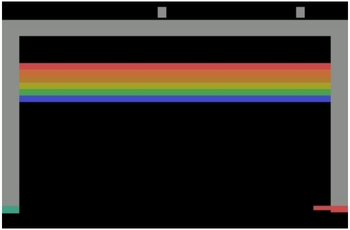
Recap of Tabular Q-Learning



- Possible solution: create a finer discretisation of states
- Now, each state represents a smaller region of continuous space
- But is this practical?

Limitation 1 of Tabular Q-Learning: Memory



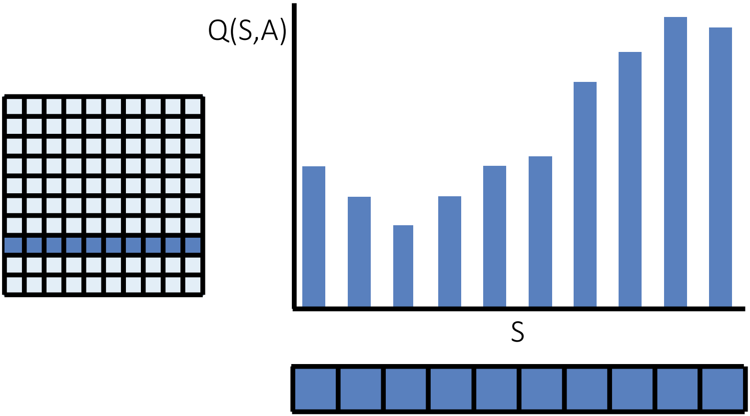


- MuJoCo robotics simulator
- 28 joints
- One degree discretisation
- Number of states $= 360^{28} \sim 10^{70}$

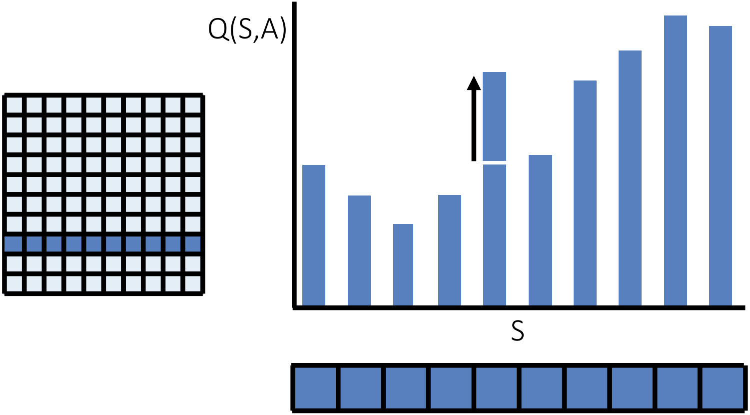
- Atari game
- 160×192 resolution
- 128 colours
- Number of states $= 128^{160\times192} \sim 10^{30,000}$

Impractical to store this in memory
(And this doesn't even include the size of the action space ...)

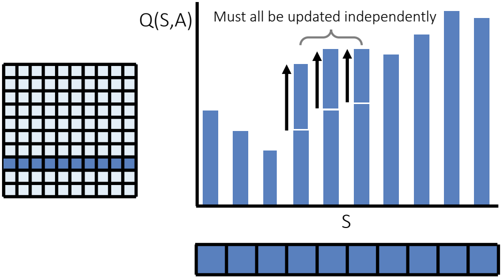
Limitation 2 of Tabular Q-Learning: Exploration



Limitation 2 of Tabular Q-Learning: Exploration

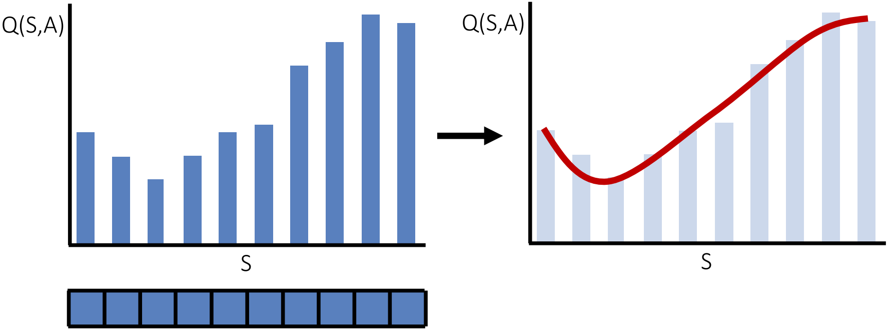


Limitation 2 of Tabular Q-Learning: Exploration



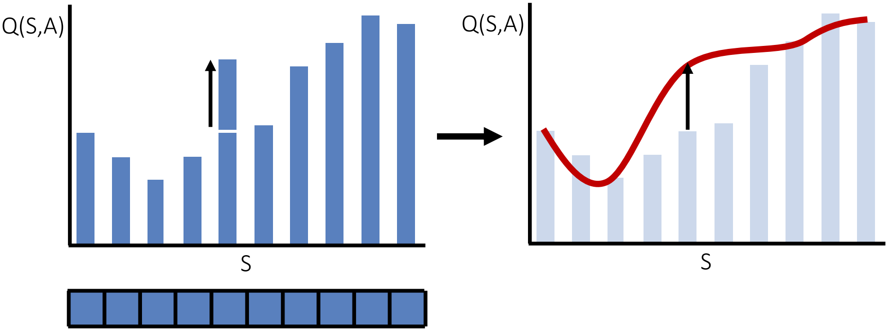
- Every state-action pair must be visited at least once to learn about its reward (and hence Q-value)
- For large state-action spaces, this makes exploration very inefficient

Solution: Function Approximation



- $Q(S, A)$ is now a continuous function, instead of a table
- Approximates the true underlying $Q(S, A)$
- If the function is parameterised as $Q_\theta(S, A)$ (e.g. a neural network), memory is constant: this solves Limitation 1

Solution: Function Approximation



- Knowledge is now "shared" (generalisation): this solves Limitation 2