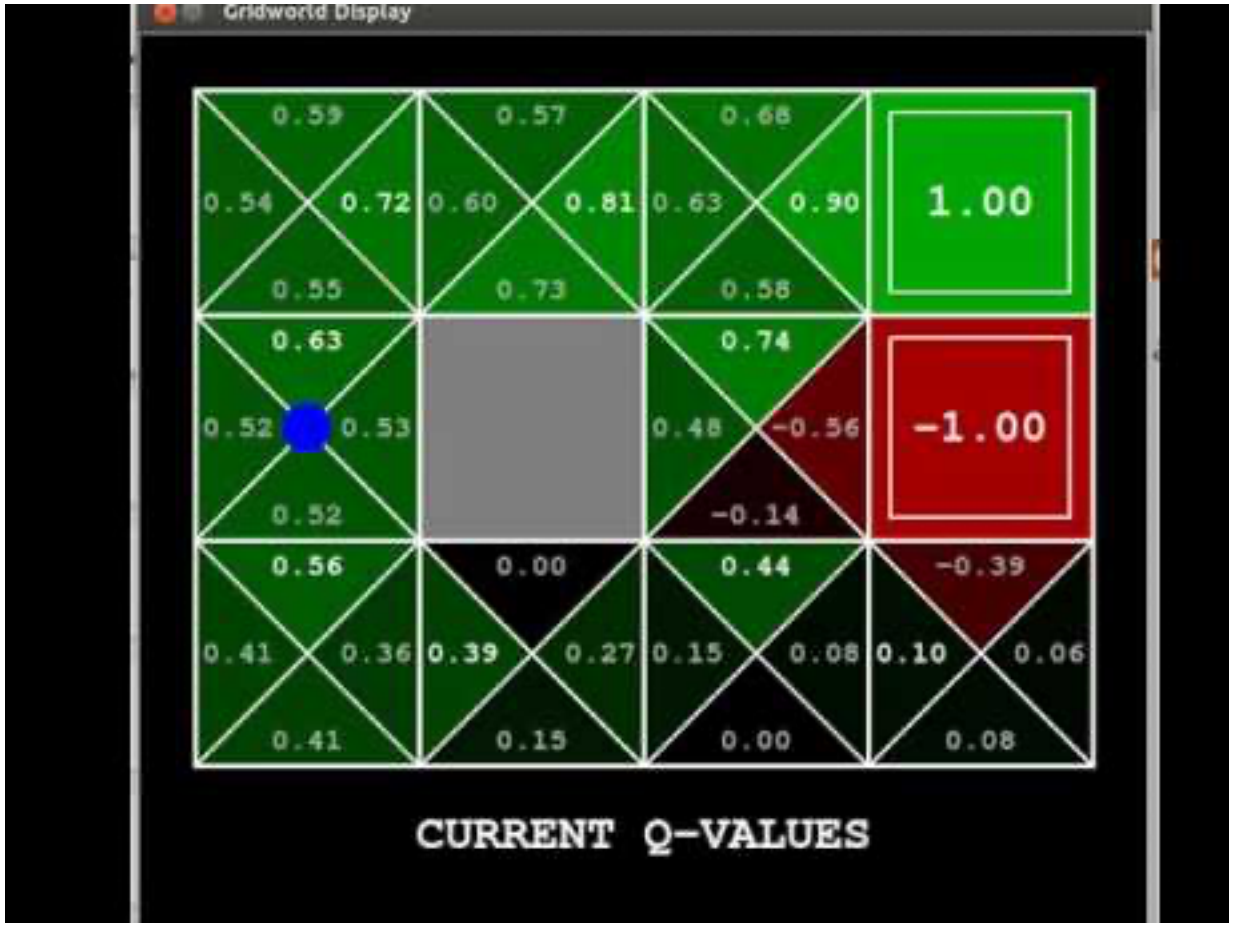
Deep Q Learning

Week 8

Recap

- Trees
  - Deterministic
  - Fully Observable
  - Search
- Markov Decision Processes
  - Probabilistic
  - Fully Observable
  - Expected Value
- POMDPs
  - Probabilistic
  - Partially Observable
  - Expected Value
- Tabular Q-Learning
  - Probabilistic
  - Expected Action-State

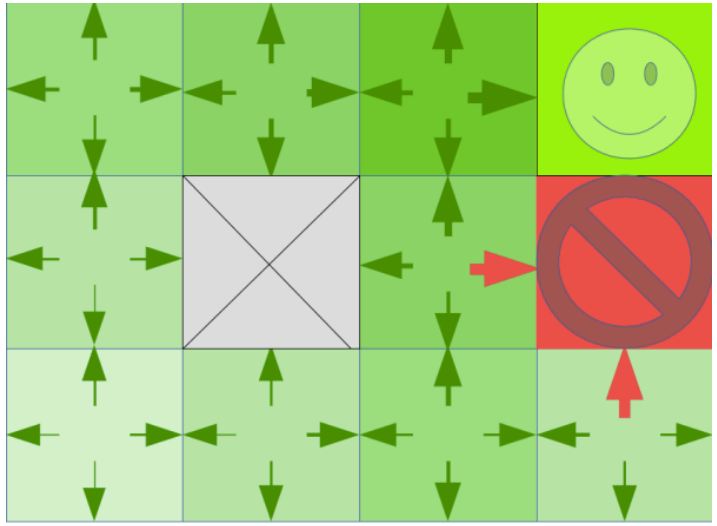
Tabular Q-Learning



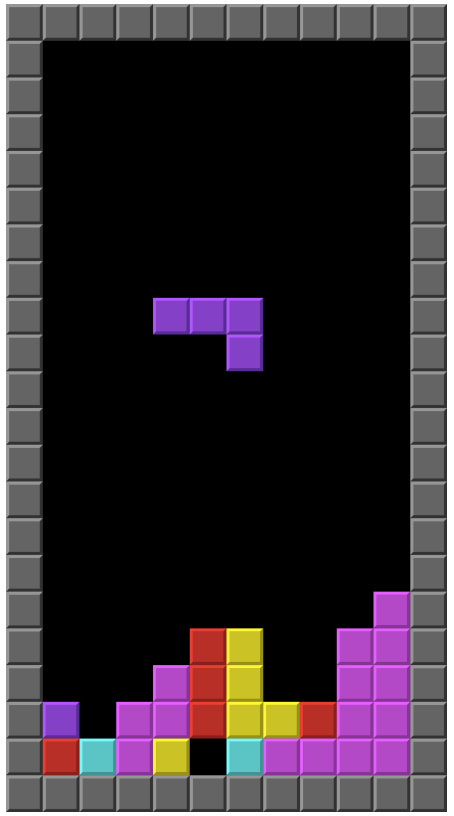
Problems with Q-Learning

- Slow!
  - We don’t get to leverage state information.
- Discrete States
- Discrete Actions

Scaling



10^{1} States



10^{60} States

Conversations?

5,000 words
15 words/sentence
10^{60} States

Continuous Spaces

Q-Function

- Q-Value represents expected reward
- Let’s just guess the value!
- Use information about state to inform our guess

Linear Function Values

- Using a feature representation, we can write a Q function (or value function) for any state using a few weights:

$$V(s) = w_1 f_1(s) + w_2 f_2(s) + \ldots + w_n f_n(s)$$

$$Q(s,a) = w_1 f_1(s,a) + w_2 f_2(s,a) + \ldots + w_n f_n(s,a)$$

- Advantage: our experience is summed up in a few powerful numbers

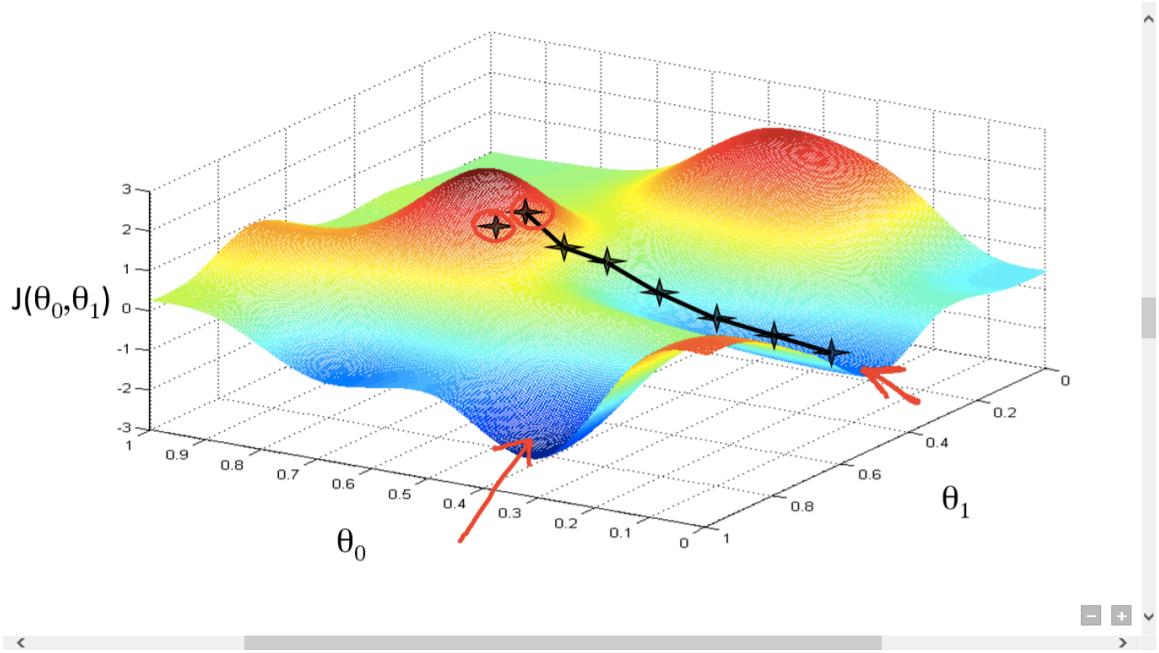
- Disadvantage: states may share features but actually be very different in value!

Approximate Q-Learning

$$Q(s,a) = w_1 f_1(s,a) + w_2 f_2(s,a) + \ldots + w_n f_n(s,a)$$

- How do we make updates?
- Gradient Descent!

$$w := w - \eta \nabla Q_i(w)$$



Stochastic Gradient Descent

$$y = w_1 + w_2 x$$

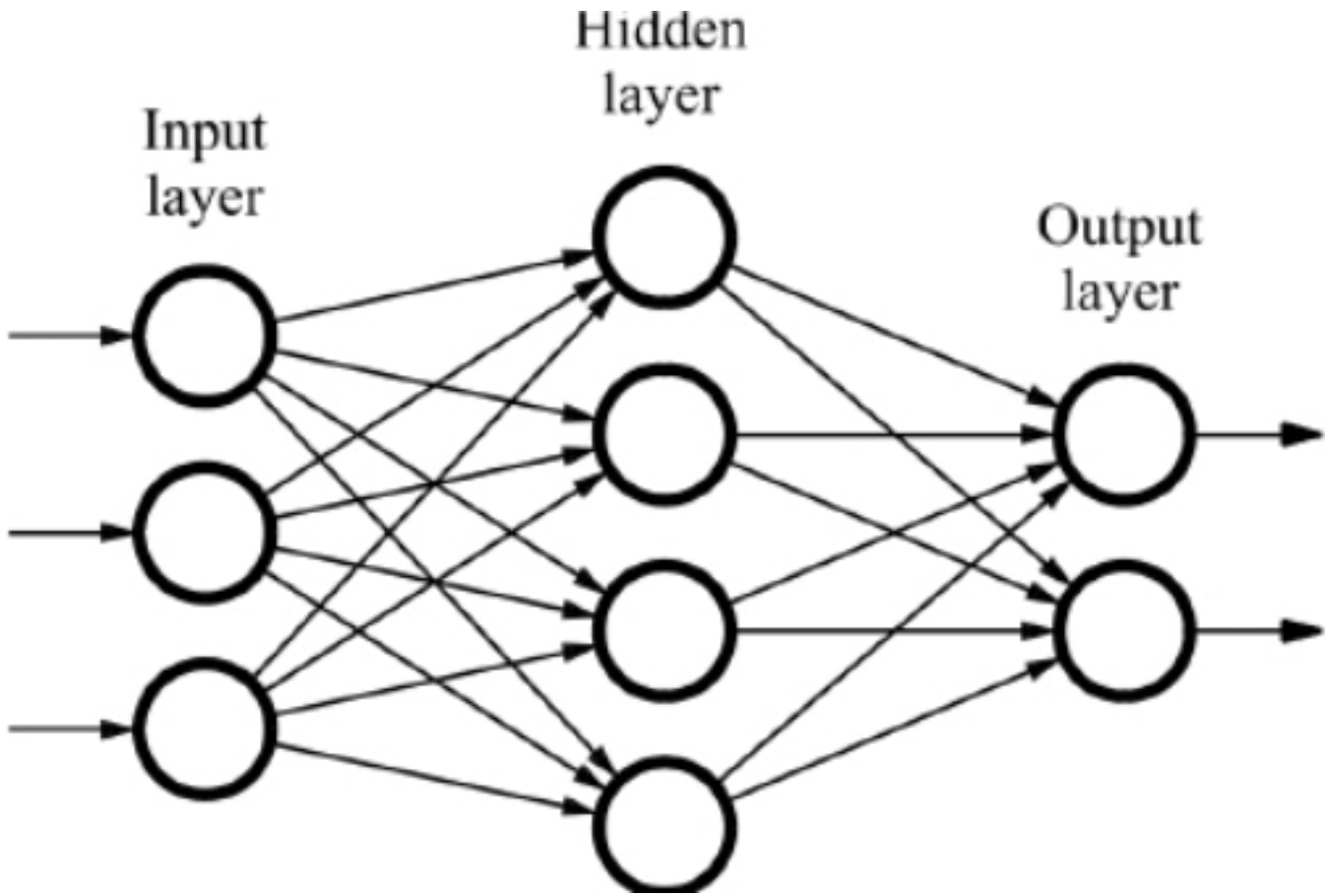
$$Q(w) = \sum_{i=1}^{n} Q_i(w) = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 = \sum_{i=1}^{n} (w_1 + w_2 x_i - y_i)^2.$$

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} := \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} - \eta \begin{bmatrix} \frac{\partial}{\partial w_1} (w_1 + w_2 x_i - y_i)^2 \\ \frac{\partial}{\partial w_2} (w_1 + w_2 x_i - y_i)^2 \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} - \eta \begin{bmatrix} 2(w_1 + w_2 x_i - y_i) \\ 2x_i (w_1 + w_2 x_i - y_i) \end{bmatrix}$$

Deep Q Learning

- We can really just use any function
- Let’s use neural networks!

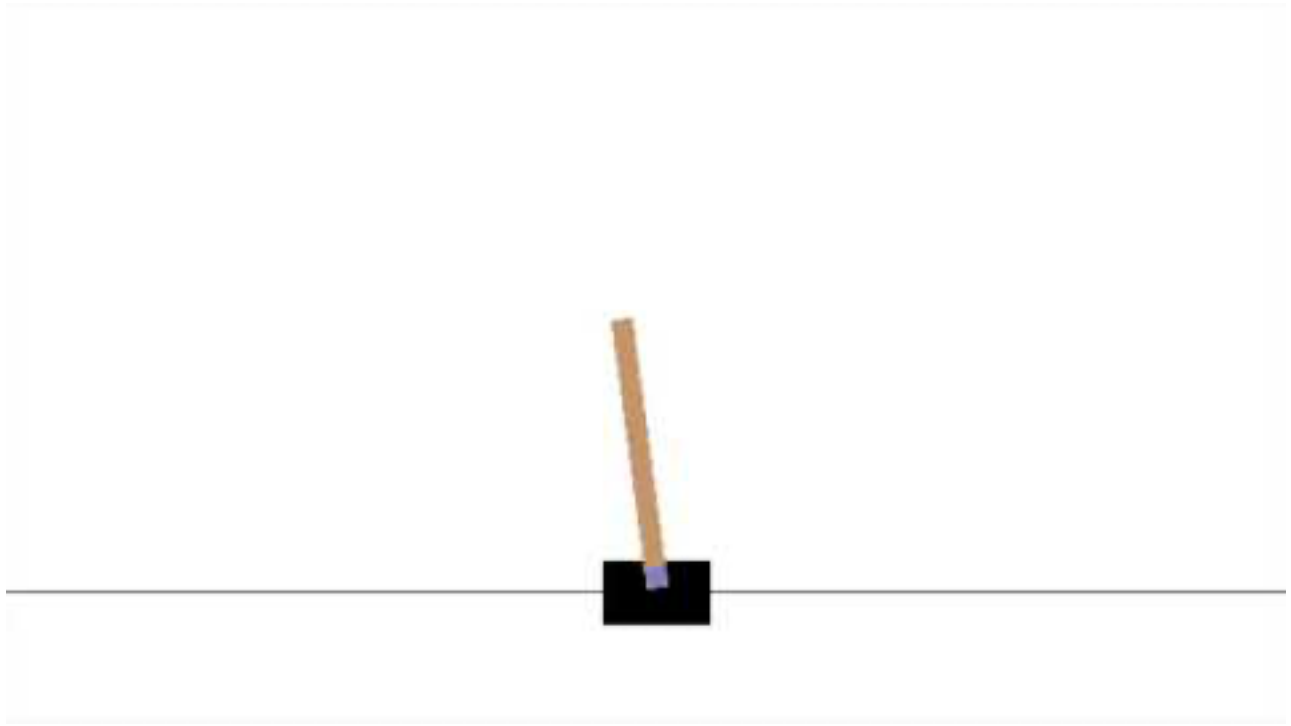
Neural Network



: $w_1 f_1(s,a) + w_2 f_2(s,a) + \ldots + w_n f_n(s,a)$

MNIST

Deep Q Learning: Cartpole



Deep Q Learning

Open up Jupyter Notebook