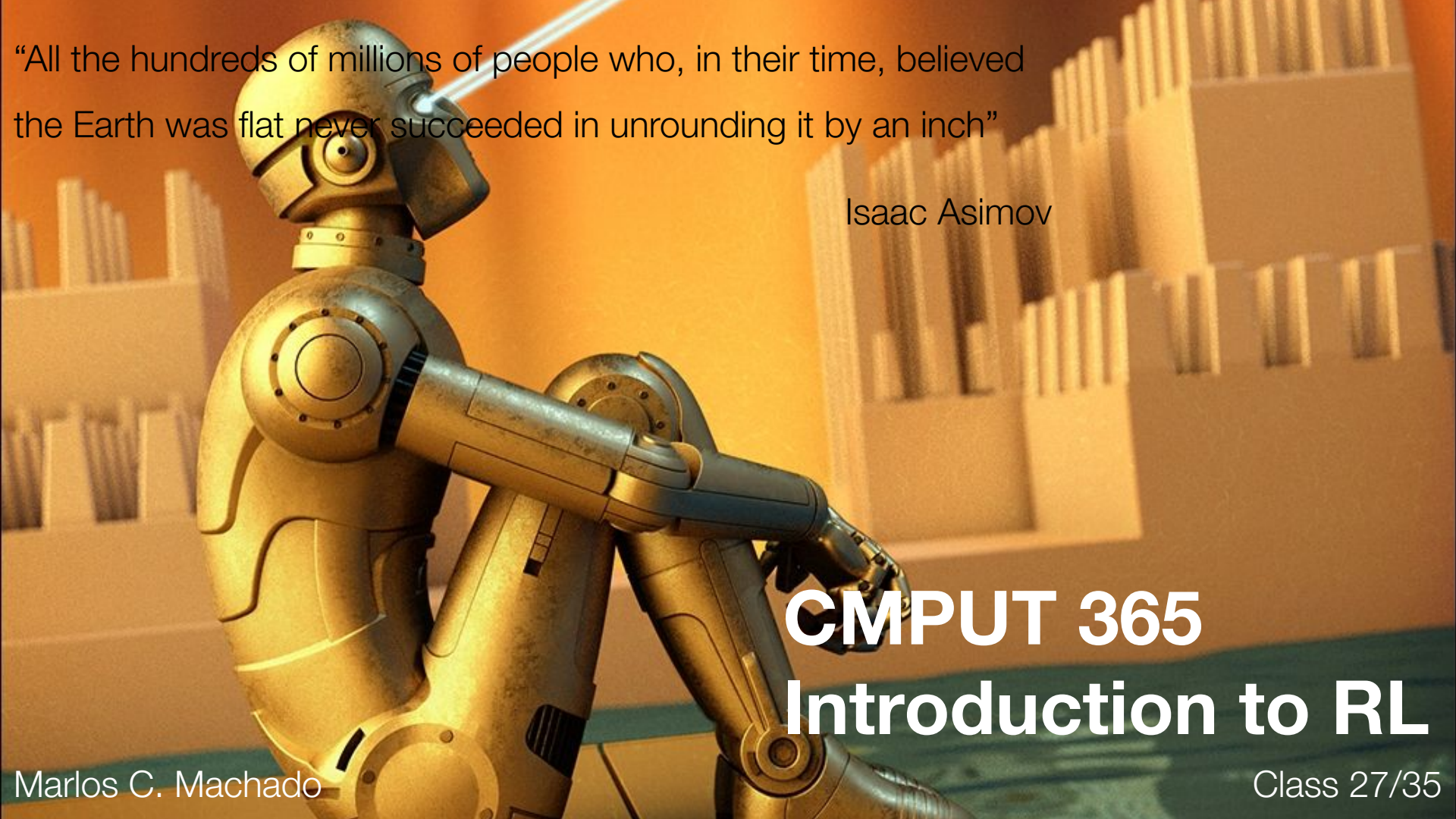

“All the hundreds of millions of people who, in their time, believed
the Earth was flat never succeeded in unrounding it by an inch”
Isaac Asimov
CMPUT 365
Introduction to RL
Marlos C. Machado
Class 27/35

Reminder

You **should be enrolled in the private session** we created in Coursera for CMPUT 365.

I **cannot** use marks from the public repository for your course marks.

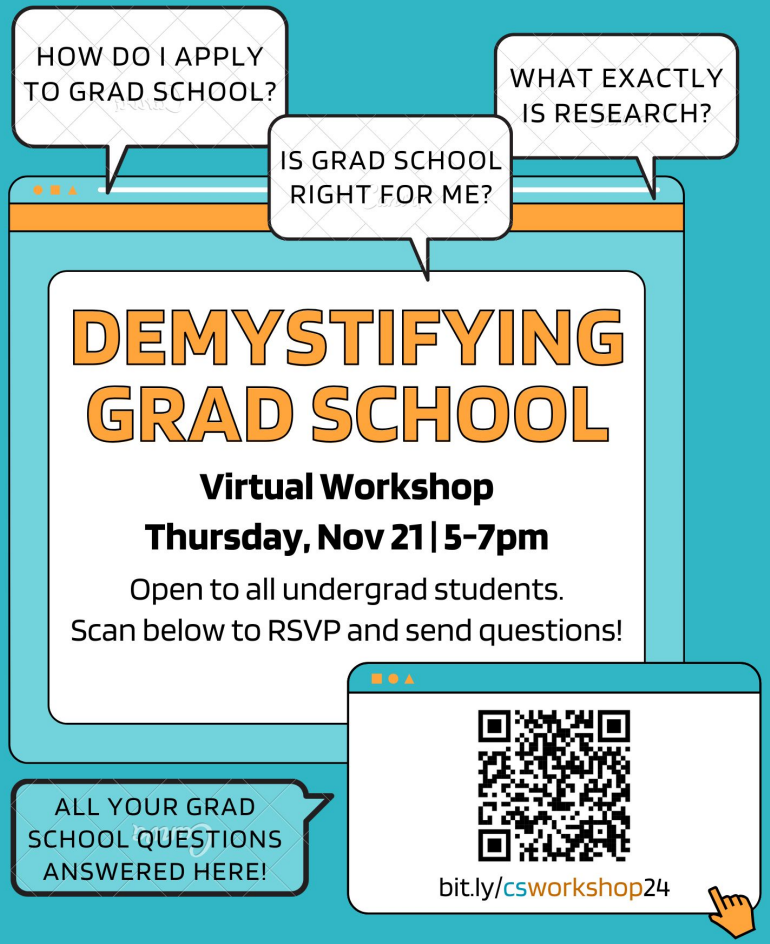
You **need** to **check**, **every time**, if you are in the private session and if you are submitting quizzes and assignments to the private section.

The deadlines in the public session **do not align** with the deadlines in Coursera.

If you have any questions or concerns, **talk with the TAs** or email us `cmput365@ualberta.ca`.

Reminders and Notes

- The programming assignment is due on Wednesday.
- Rich Sutton will give a guest lecture Dec 9th, Monday. Spread the word.
- A note on the final exam:
  - The required reading from the syllabus does not mean that's what will be covered in the final exam. There are some mismatches. Anything we discussed in class is fair game, including Maximization Bias and Double Learning (Section 6.7), and Nonlinear Function Approximation: Artificial Neural Networks (Section 9.7).



RSVP form (not required, but appreciated):

https://docs.google.com/forms/d/11odJJgO3kgJ_XFDq9vnEz4FABjlNKx7-iL6AkCJ67ZQ/edit

Direct link to the zoom:

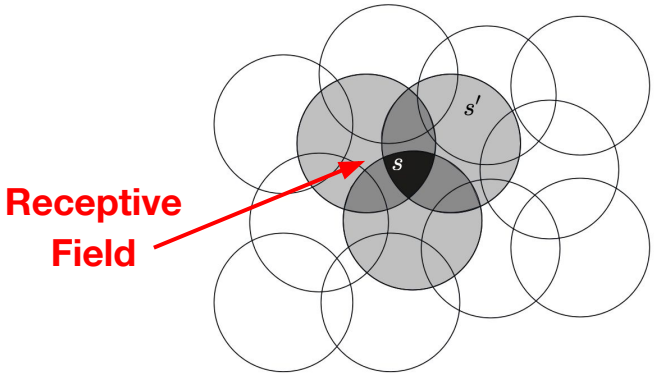
https://ualberta-ca.zoom.us/j/93282952849?pwd=eqE7hm46hwMJS02EZoqjw5GOngtWkK.1

Please, interrupt me at any time!

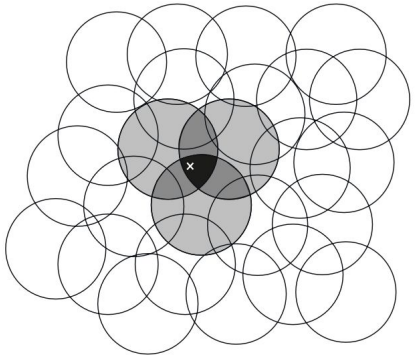
https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Last Class: Coarse Coding

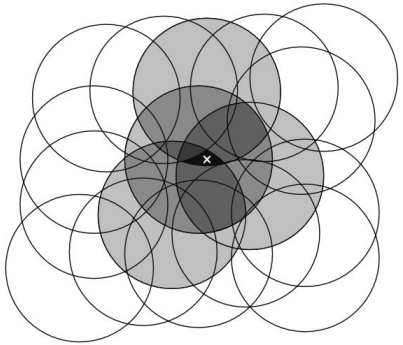
- Consider a task in which the natural representation of the state set is a continuous two- dimensional space.
- We define binary features indicating whether a state is present or not in a specific circle.



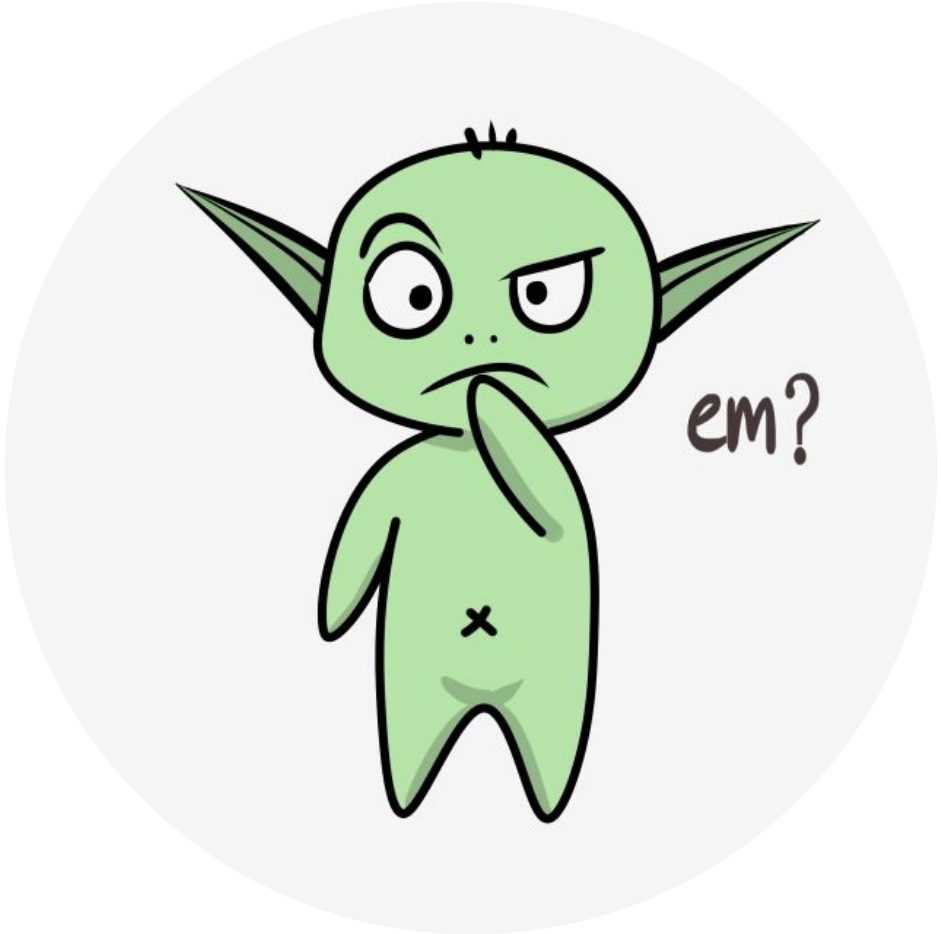
The shape defines generalization



Narrow generalization



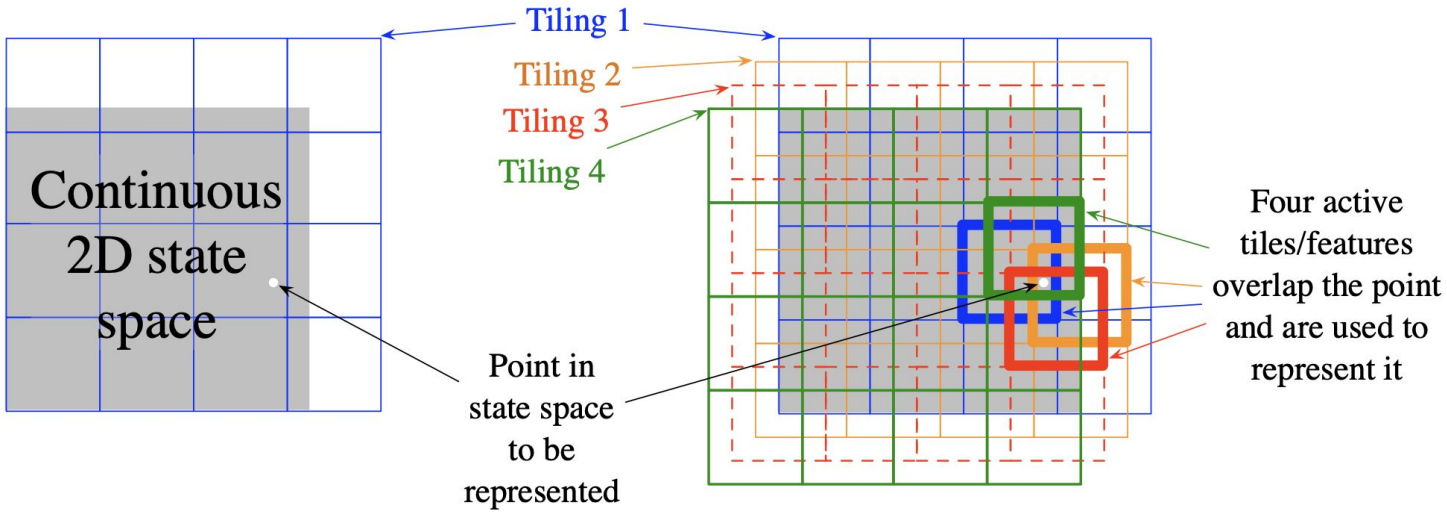
Broad generalization



https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

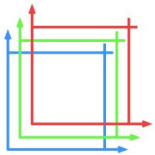
Tile Coding

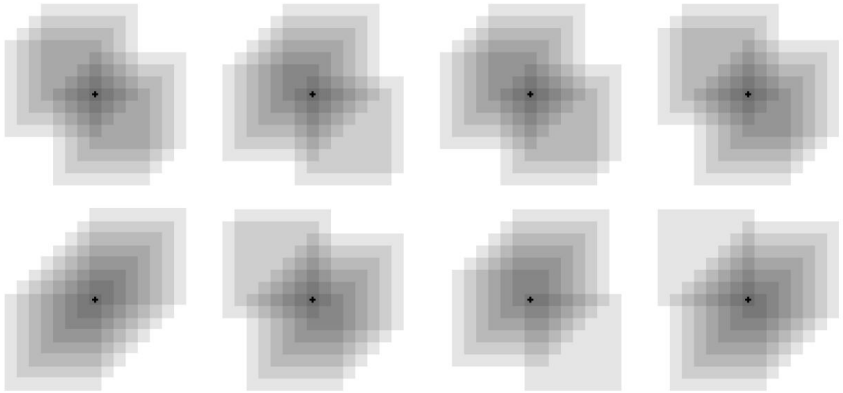
- Tile coding is a form of coarse coding for multi-dimensional continuous spaces (with a fixed number of active features per timestep).



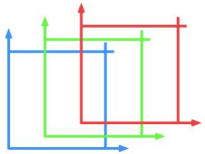
Tile Coding

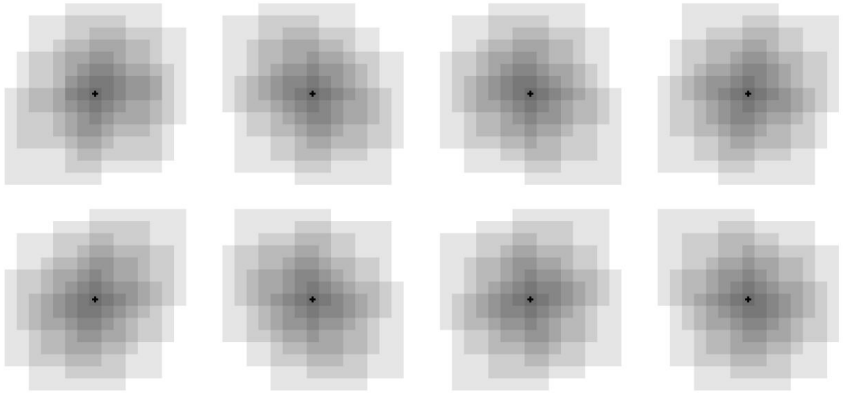
Possible generalizations for uniformly offset tilings

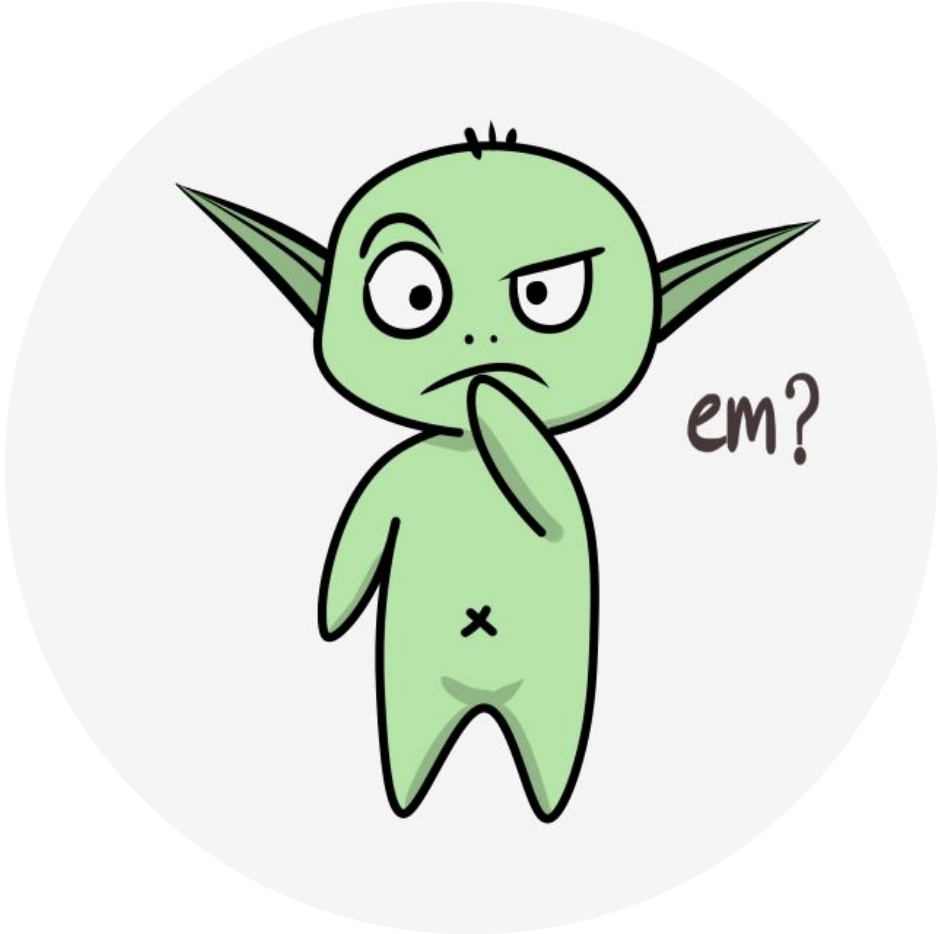




Possible generalizations for asymmetrically offset tilings





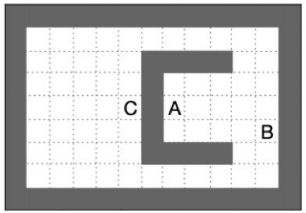


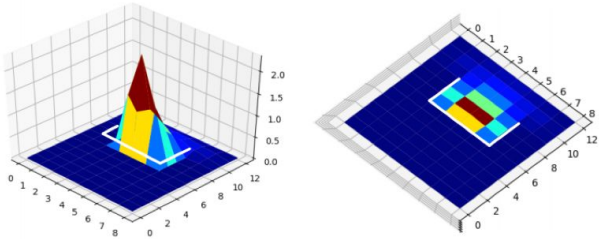
https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

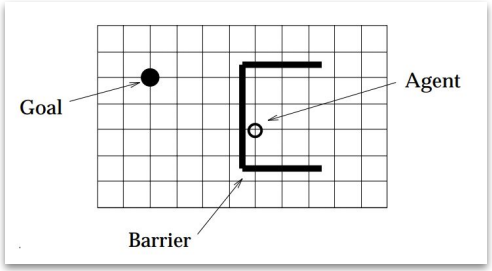
It Isn’t that We do Function Approximation Because We Cannot do Tabular Reinforcement Learning

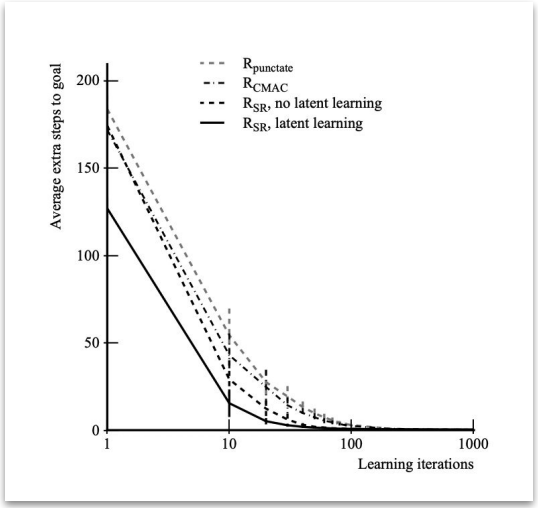
- Successor Representation [Dayan, Neural Computation 1993].

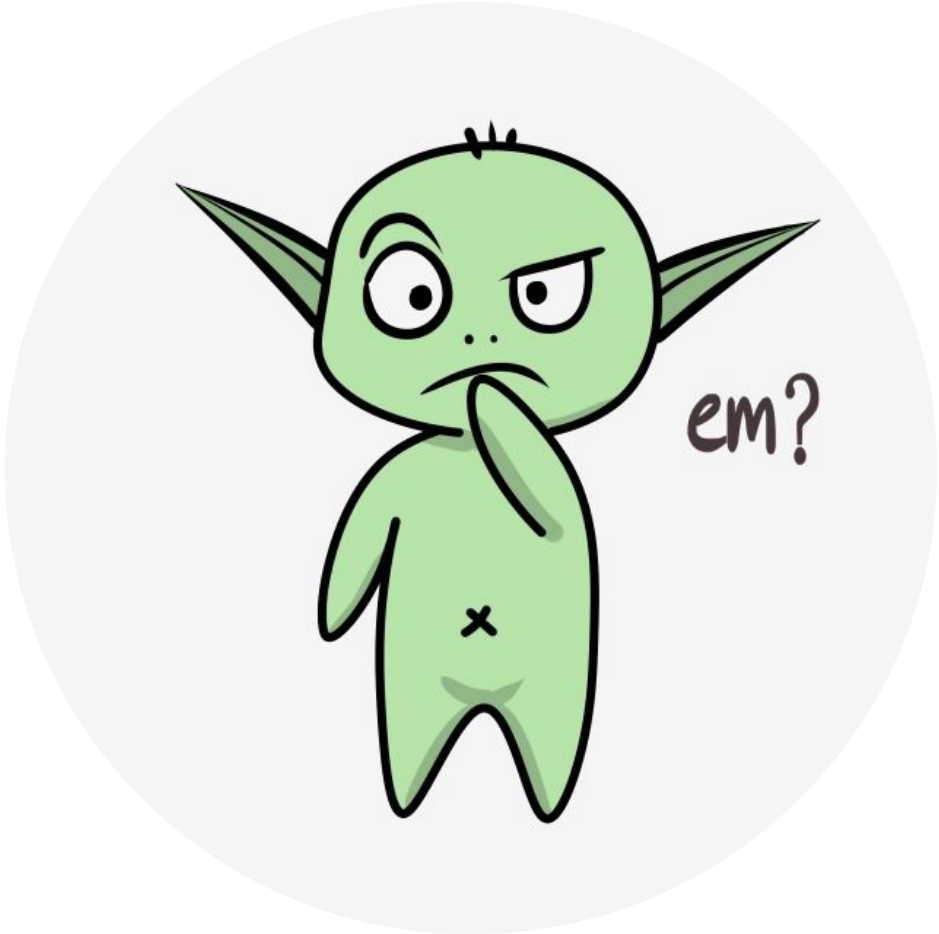
$$\Psi_\pi(s, s') = \mathbb{E}_\pi \left[\sum_t \gamma^t \mathbf{1}_{S_t = s'} | S_0 = s \right]$$









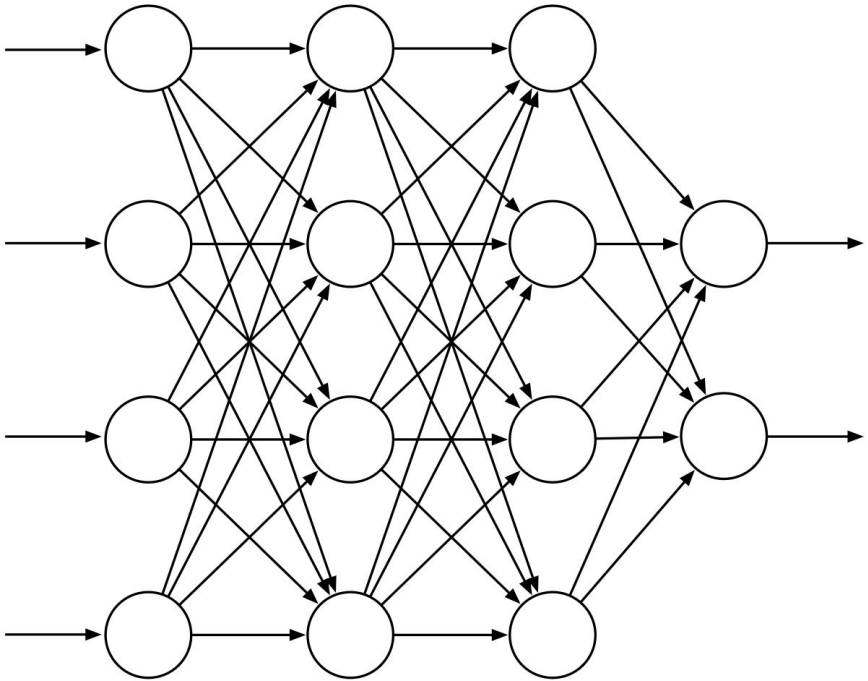


https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

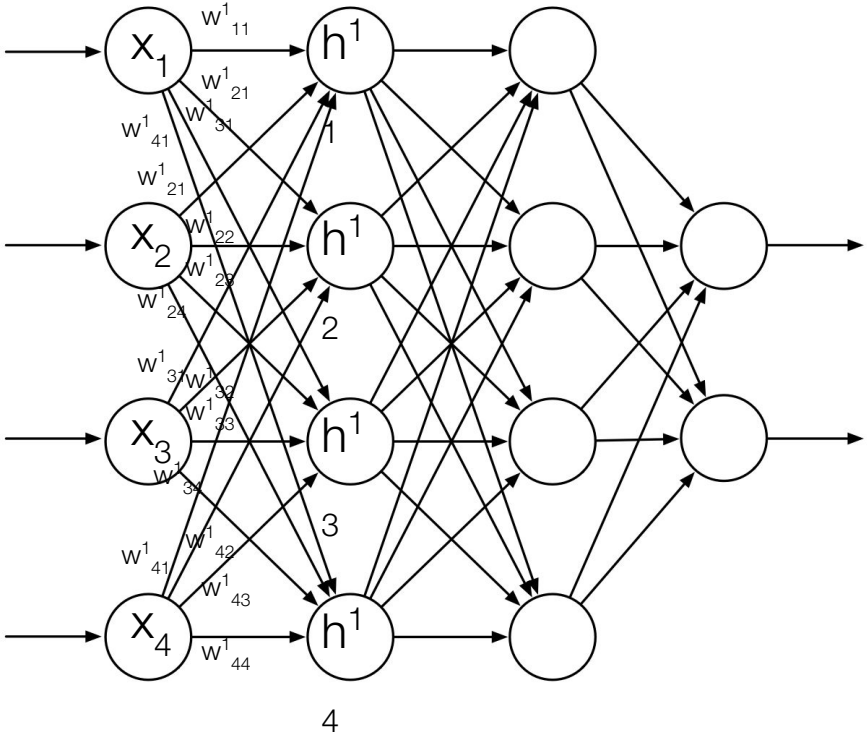
Nonlinear Function Approximation: Artificial Neural Networks

- The basics of deep reinforcement learning.
- Idea: Instead of using linear features, we feed the “raw” input to a neural network and ask it to predict the state (or state-action) value function.

Neural Networks



Neural Networks



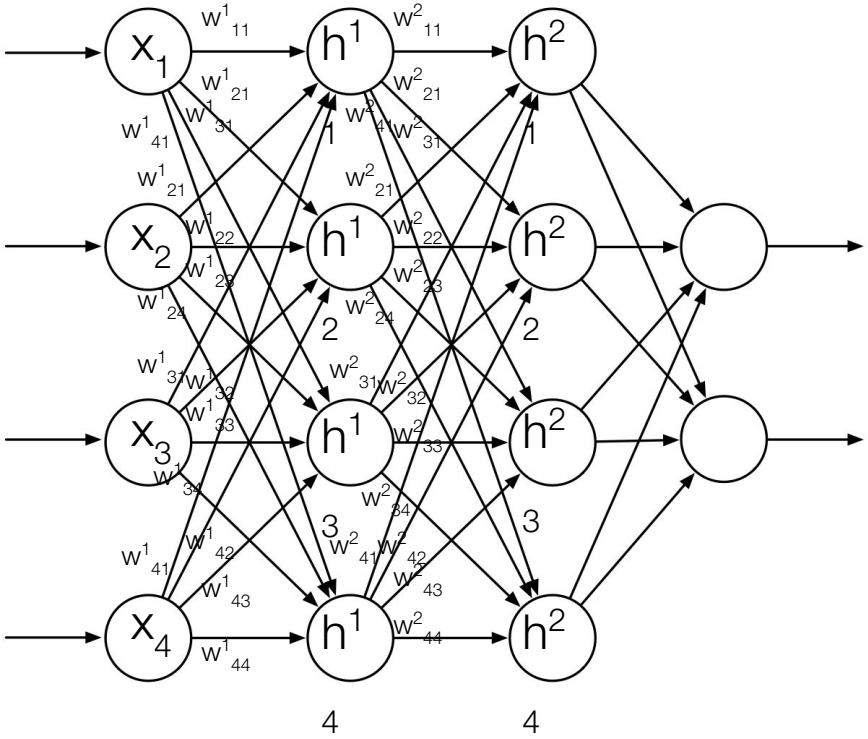
The activation function introduces non-linearity

E.g.: *f(x) = max(0, x)*

$\mathbf{h^1} = \text{act}(\mathbf{xW^1} + \mathbf{b^1})$

s.t. $h^1_1 = x_1 w^1_{11} + x_2 w^1_{21} + x_3 w^1_{31} + x_4 w^1_{41} + B^1$

Neural Networks



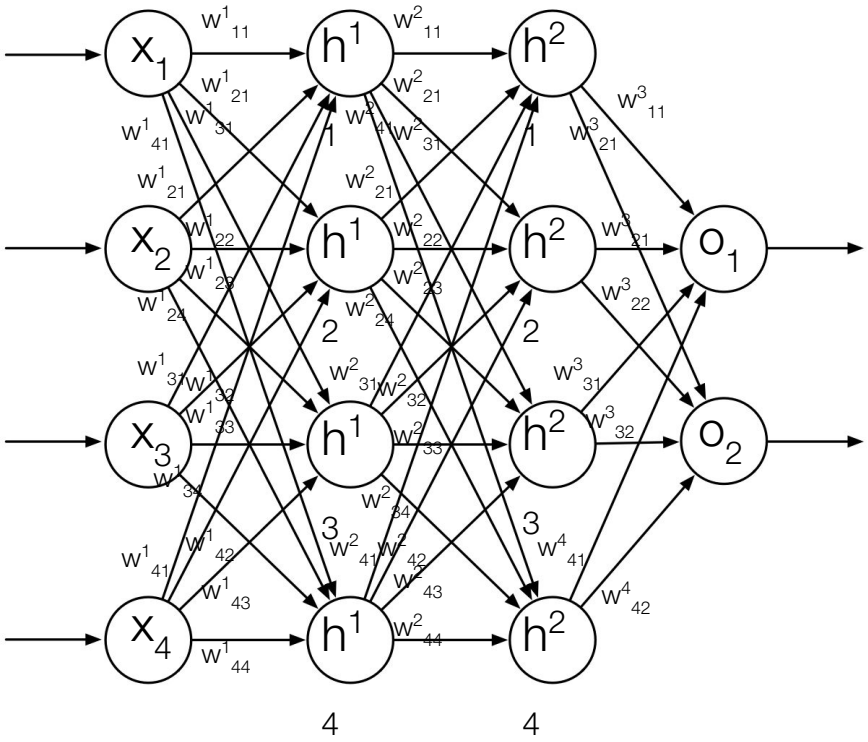
$\mathbf{h^1} = \text{act}(\mathbf{xW^1} + \mathbf{b^1})$

s.t. $h^1_1 = x_1 w^1_{11} + x_2 w^1_{21} + x_3 w^1_{31} + x_4 w^1_{41} + B^1$

$\mathbf{h^2} = \text{act}(\mathbf{h^1 W^2} + \mathbf{b^2})$

s.t. $h^2_1 = h^1_1 w^2_{11} + h^1_2 w^2_{21} + h^1_3 w^2_{31} + h^1_4 w^2_{41} + B^2$

Neural Networks



$\mathbf{h^1} = \text{act}(\mathbf{xW^1} + \mathbf{b^1})$

s.t. $h^1_1 = x_1 w^1_{11} + x_2 w^1_{21} + x_3 w^1_{31} + x_4 w^1_{41} + B^1$

$\mathbf{h^2} = \text{act}(\mathbf{h^1 W^2} + \mathbf{b^2})$

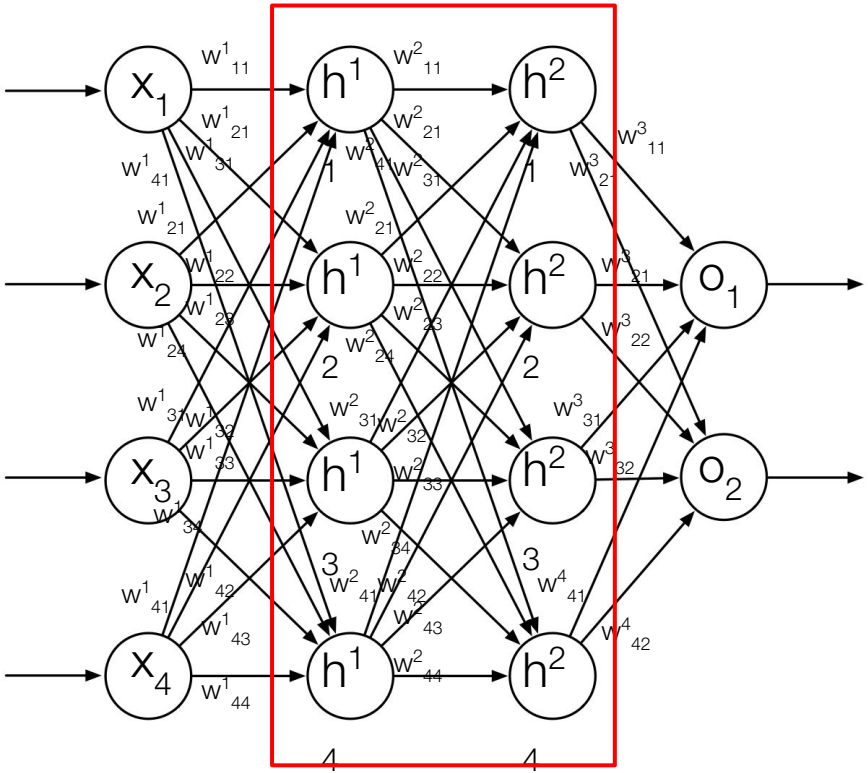
s.t. $h^2_1 = h^1_1 w^2_{11} + h^1_2 w^2_{21} + h^1_3 w^2_{31} + h^1_4 w^2_{41} + B^2$

$\mathbf{o} = \text{act}(\mathbf{h^2 W^3} + \mathbf{b^3})$

s.t. $o_1 = h^2_1 w^3_{11} + h^2_2 w^3_{21} + h^2_3 w^3_{31} + h^2_4 w^3_{41} + B^2$

$\mathbf{o} = \text{act}(\text{act}(\text{act}(\mathbf{xW^1} + \mathbf{b^1})\mathbf{W^2} + \mathbf{b^2})\mathbf{W^3} + \mathbf{b^3})$

Neural Networks



$\mathbf{h^1} = act(\mathbf{xW^1} + \mathbf{b^1})$

s.t. $h^1_1 = x_1 w^1_{11} + x_2 w^1_{21} + x_3 w^1_{31} + x_4 w^1_{41} + B^1$

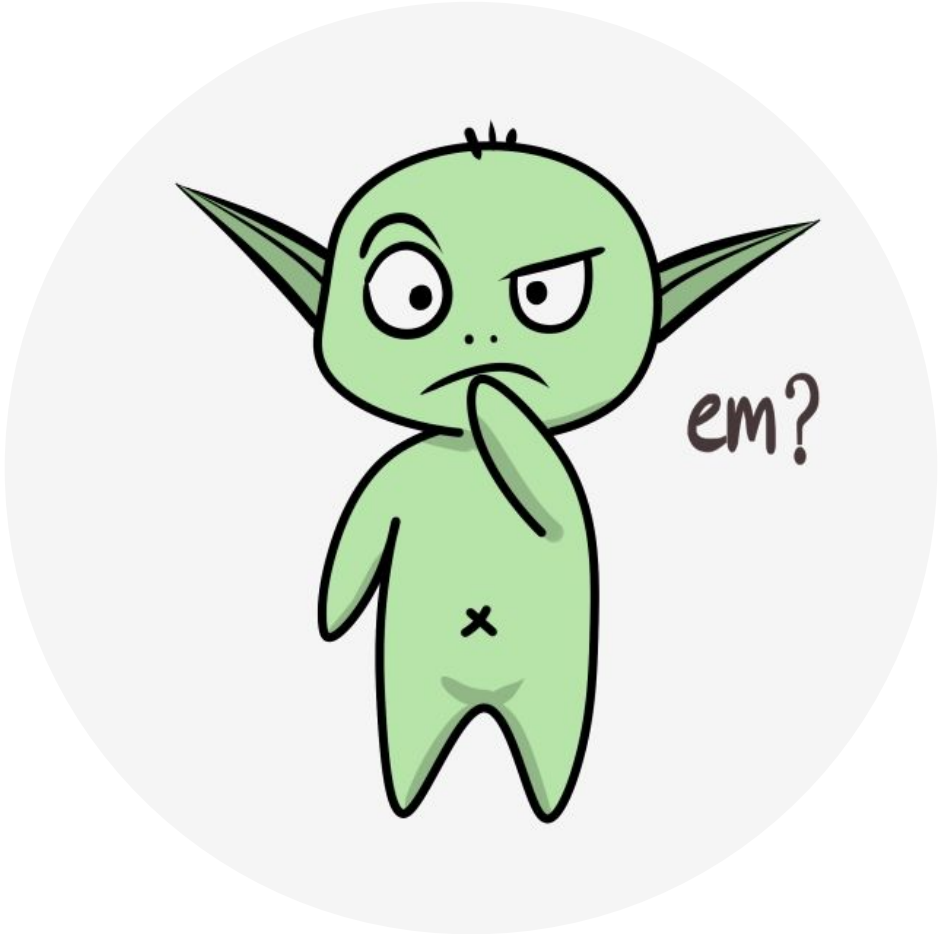
$\mathbf{h^2} = act(\mathbf{h^1 W^2} + \mathbf{b^2})$

s.t. $h^2_1 = h^1_1 w^2_{11} + h^1_2 w^2_{21} + h^1_3 w^2_{31} + h^1_4 w^2_{41} + B^2$

$\mathbf{o} = act(\mathbf{h^2 W^3} + \mathbf{b^3})$

s.t. $o_1 = h^2_1 w^3_{11} + h^2_2 w^3_{21} + h^2_3 w^3_{31} + h^2_4 w^3_{41} + B^2$

$\mathbf{o} = act(act(act(\mathbf{xW^1} + \mathbf{b^1})\mathbf{W^2} + \mathbf{b^2})\mathbf{W^3} + \mathbf{b^3})$



https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht