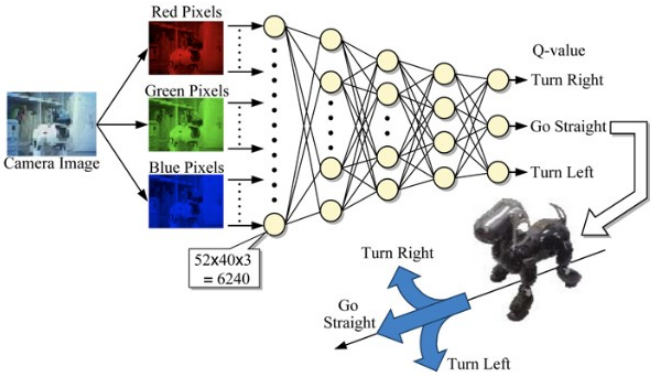
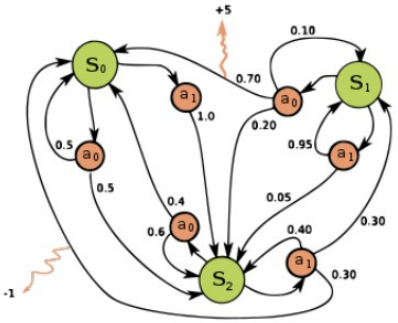
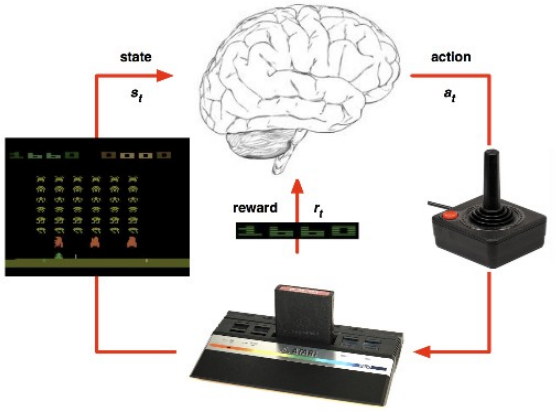
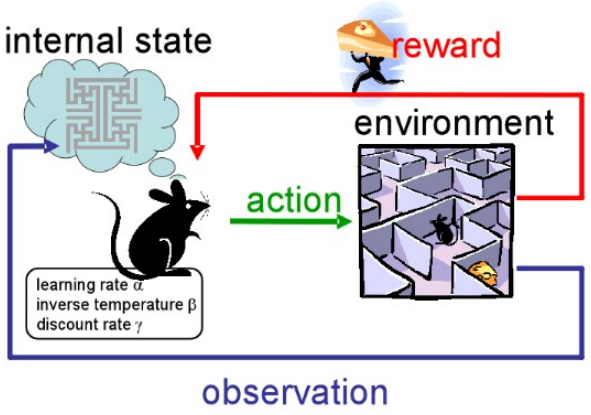
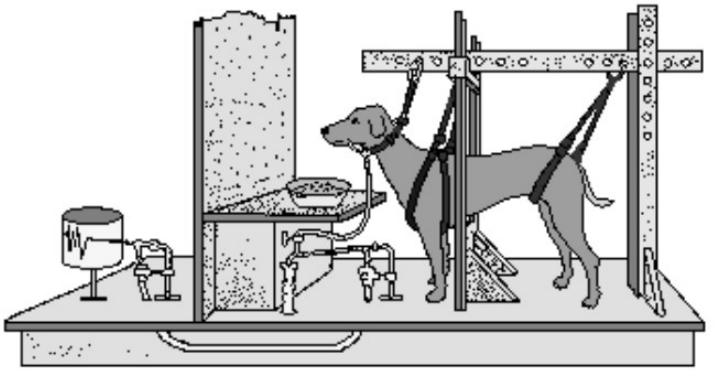
COMP 138: Reinforcement Learning











Instructor: Jivko Sinapov

Announcements

Reading Assignment

- Chapter 10 of SB
- Mnih, Volodymyr, et al. "Playing atari with deep reinforcement learning." arXiv preprint arXiv:1312.5602 (2013).
- Responses should discuss both readings
- You get extra credit for answering others' questions!

Why Function Approximation

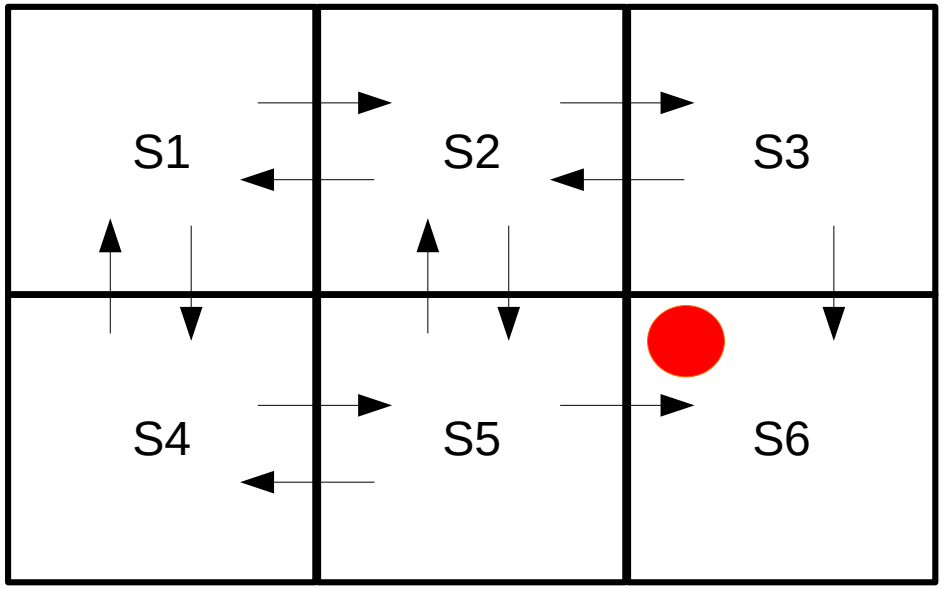
Exercise (white board)

Moderated Discussion

Today: RL with Function Approximation

$$Q(s,a) = \sum_{i=1}^{n} f_i(s,a) w_i$$

The limitations of Tabular Methods



+ 100 reward for getting to S6
0 for all other transitions

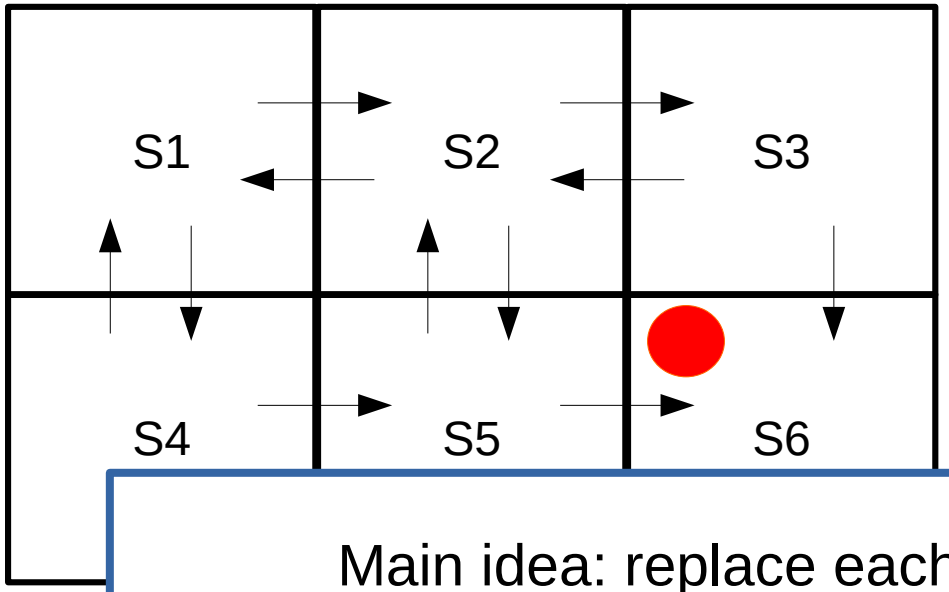
Update rule upon executing action a, ending up in state s' and observing reward r :

$$Q(s , a)=r +\gamma \max a' Q(s' ,a ')$$

γ = 0.5 (discount factor)

Q-Table

S1	right	25
S1	down	25
S2	right	50
S2	left	12.5
S2	down	50
S3	left	25
S3	down	100
S4	up	12.5
S4	right	50
S5	left	25
S5	up	25
S5	right	100



Main idea: replace each state-action pair with a feature vector

+ 1
0 for all other transitions

Q-Table

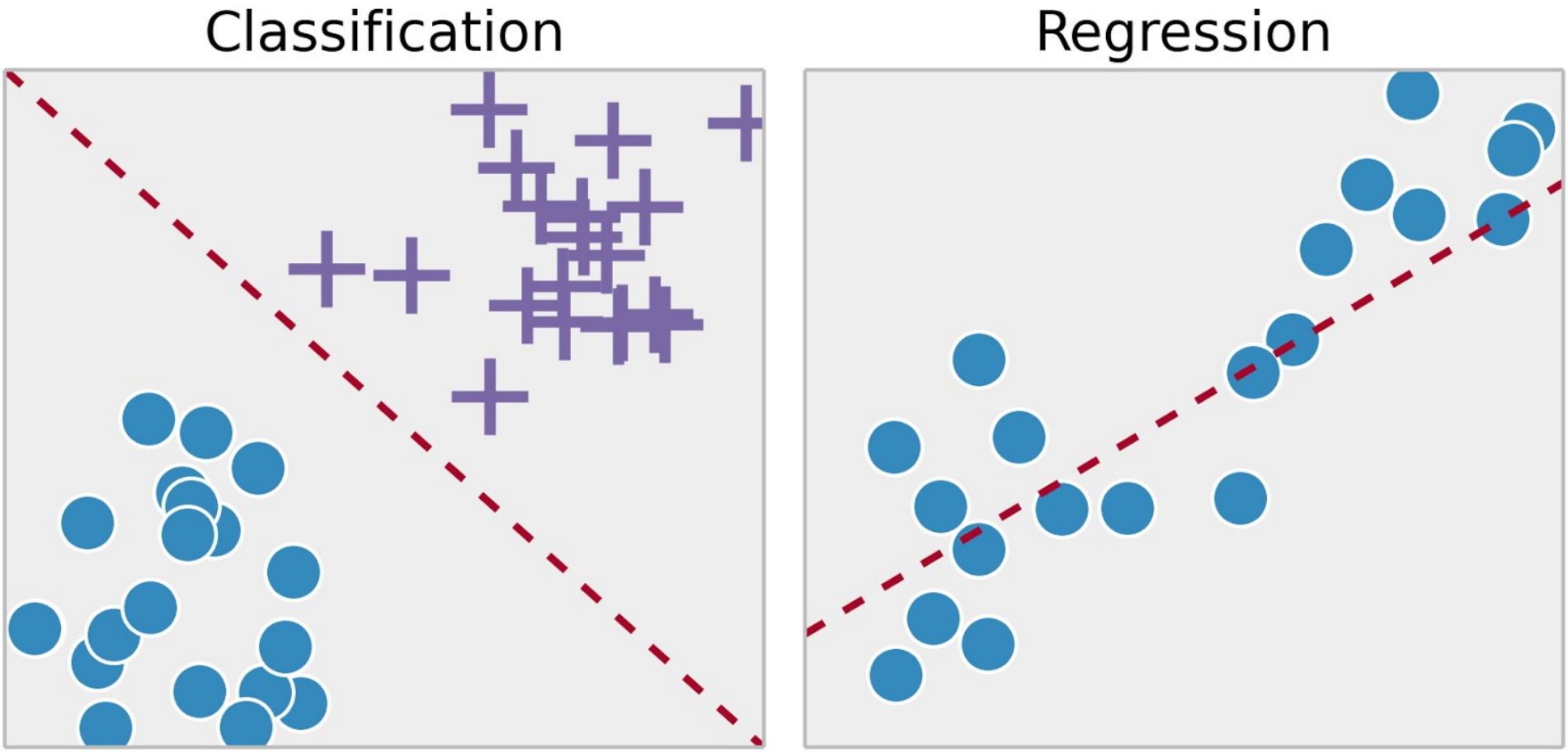
S1	right	25
S1	down	25
S2	right	50
S2	left	12.5
S2	down	50
S3	left	25
0.5, -0.7, 0.2, …, 0.9		100
S4	up	12.5
S4	right	50
S5	left	25
S5	up	25
S5	right	100

Update rule upon executing action a, ending up in state s' and observing reward r :

$$Q(s, a) = r + \gamma \max_{a'} Q(s', a')$$

$\gamma = 0.5$ (discount factor)

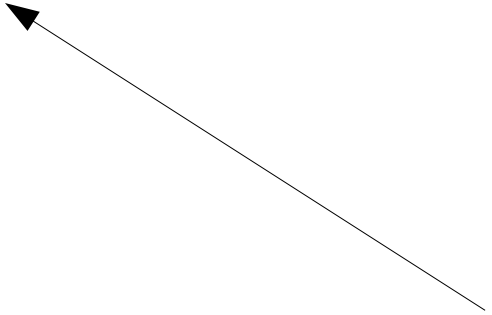
Connection to Supervised ML



[https://cdn-images-1.medium.com/max/1600/1*ASYpFfDh7XnreU-ygqXonw.png]

Linear Q-Function Approximation

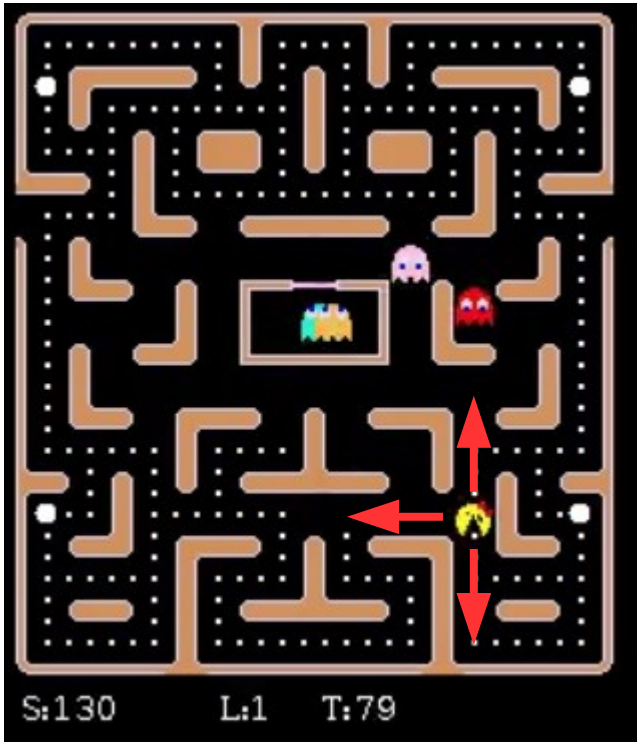
$$Q^*(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s'} \mathcal{P}(s'|s,a) \max_{a'} Q^*(s',a')$$



$$w_1 * x_1 + w_2 * x_2 + \ldots + w_n * x_n$$

$$Q(s,a) = \sum_{i=1}^{n} f_i(s,a) w_i$$

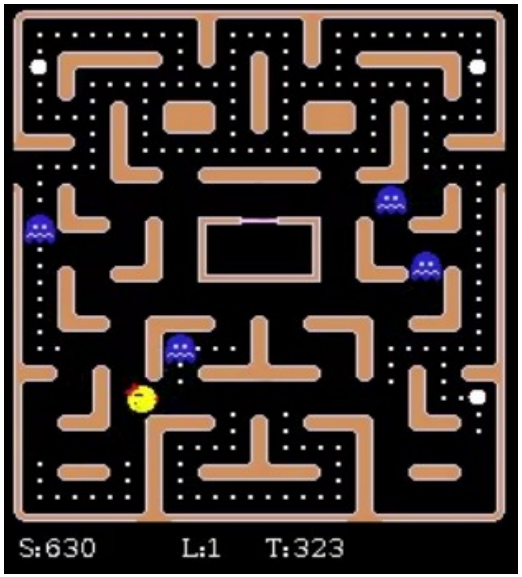
Example: Ms. Pac-man

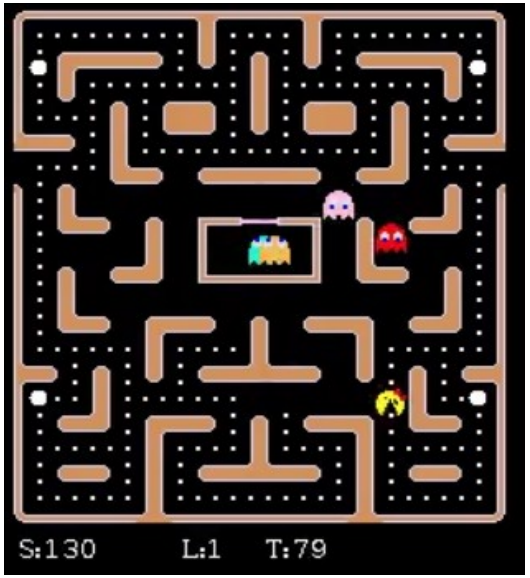


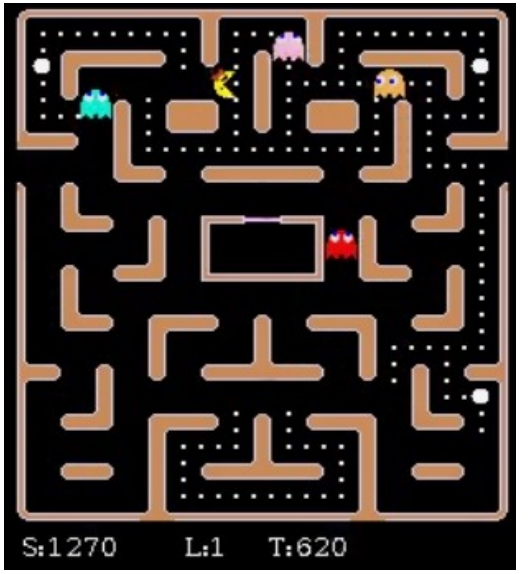
The problem: for a given action and the current configuration, compute a fixed-length feature vector

Each feature must have some semantic “meaning”

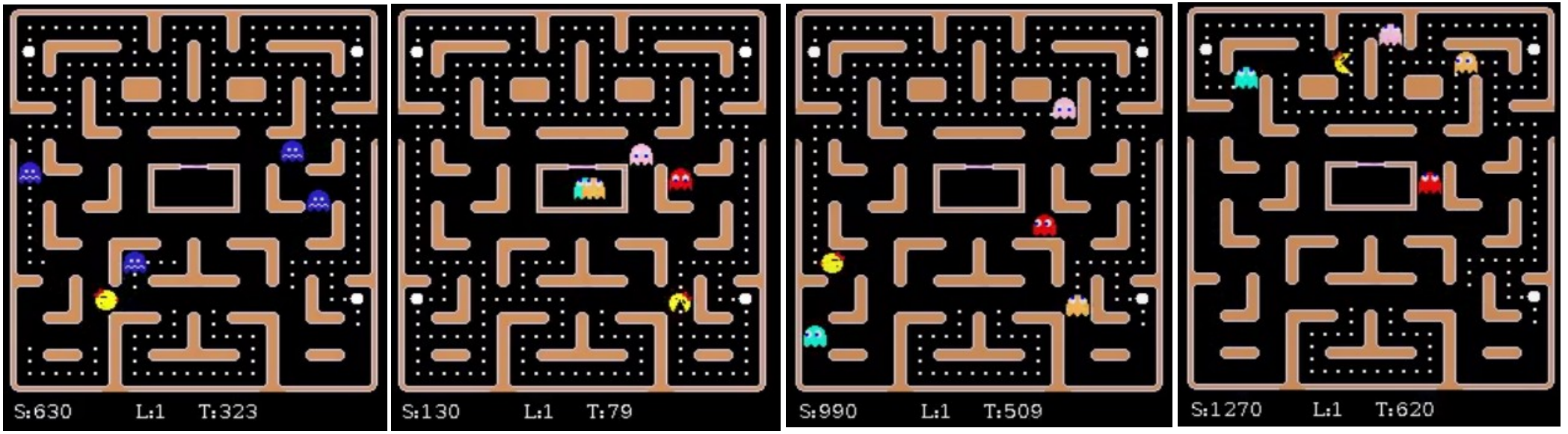
Example Configurations







Small group activity: feature engineering



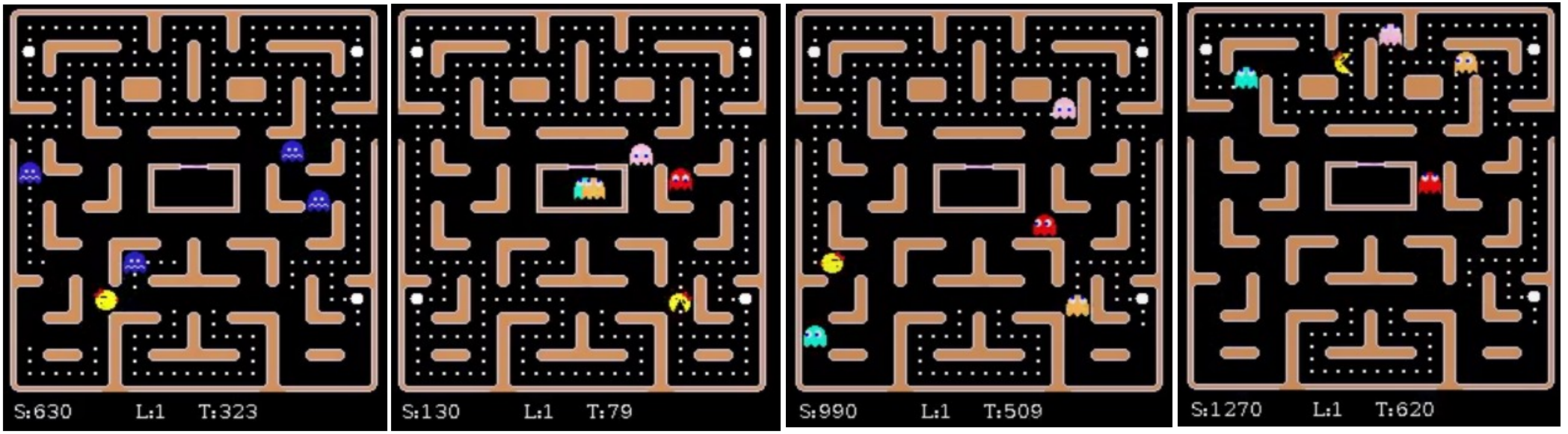
Be the feature engineer: given a configuration and a cardinal direction, design the feature types that describe how the world “looks like” in that direction; assume you have access to the underlying game simulator; the board itself is a graph with nodes and edges and for each node, you know whether there is a pill, power, pill, a ghost, and its state (edible or not, direction of movement)

Example feature: $x_{ghost\text{-}k}$ = 0.0 if no ghost is present up to *K* nodes towards the action’s direction and 1.0 otherwise

Be as precise as possible!

Assume linear q-function approximation – can you come up with an initial set of weights given the semantics of the features you designed?

Discussion – what did you come up with?



Be the feature engineer: given a configuration and a cardinal direction, design the feature types that describe how the world “looks like” in that direction; assume you have access to the underlying game simulator; the board itself is a graph with nodes and edges and for each node, you know whether there is a pill, power, pill, a ghost, and its state (edible or not, direction of movement)

Example feature: $x_{ghost\text{-}k}$ = 0.0 if no ghost is present up to *K* nodes towards the direction and 1.0 otherwise

Be as precise as possible!

Assume linear q-function approximation – can you come up with an initial set of weights given the semantics of the features you designed?

Overview of 9.1-9.3

“Local Views” for Function Approximation

Mid-term Course Evaluations

- Please list 3 things about this course that enhance your learning
- Please list 3 areas that could improve your learning in this course
- What could students in the course do to make the course better for the class and the instructor?

THE END