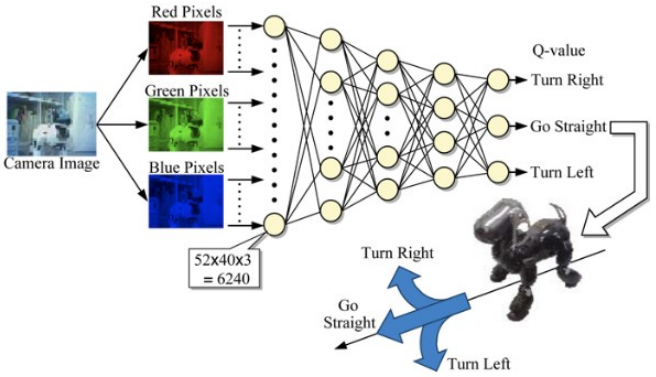
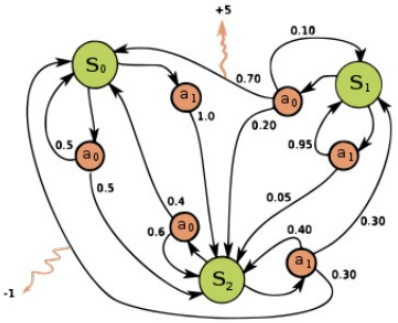
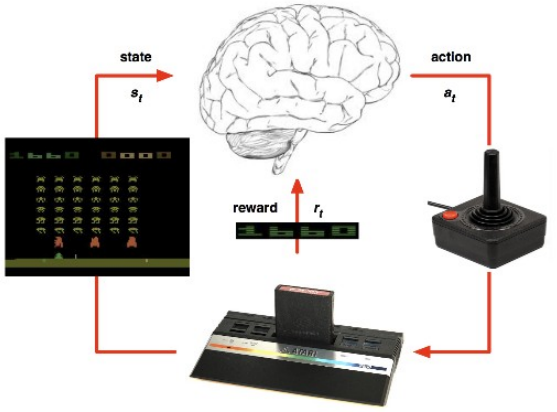
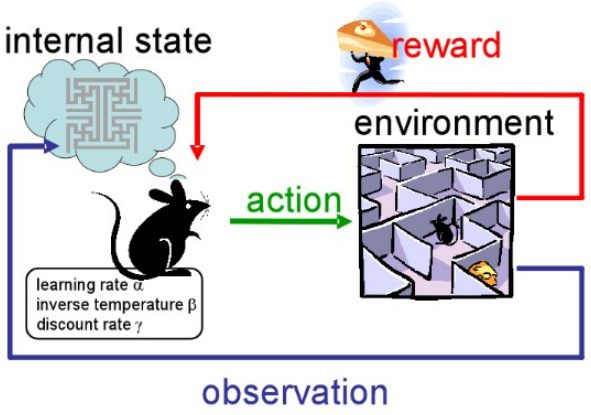
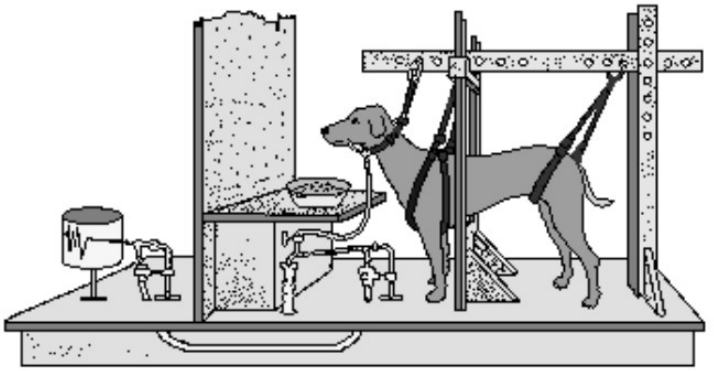
COMP 138: Reinforcement Learning











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)

Reading Discussion

“I see the weight updates, and I get the full algorithm, but it doesn't explicitly show where the weights are being used. The value function doesn't come up, and it gives an assumed policy. Where does the policy come from? I assume it's based on the linear approximation of the value function with respect to actions, but how specifically? ”

- Hayley

Reading Discussion

“For the mean squared error formula, how do we know the true values of the states for each state in the environment?”

- Channi

Reading Discussion

“In the introduction to chapter 9, the authors make a point of noting that the weights used in estimating the value function in function approximation are typically fewer than the number of states. Is this an empirical observation (from practice) or a directly imposed constraint?”

- Channi

Reading Discussion

“[Question] The textbook says on page 201 "Remember that we do not seek or expect to find a value function that has zero error for all states, but only an approximation that balances the errors in different states. If we completely corrected each example in one step, then we would not find such a balance.", is this saying that we want to avoid the overfitting problem?”

- Beier

Reading Discussion

“It was not super clear to me in which cases each technique for approximating linear functions is used. Are there general rules that dictate when each one is used?”

- Andrew

Reading Discussion

“Can one combine the concepts of coarse and tile coding? For example, in a continuous 3D state space, drawing a bunch of overlapping spheres and storing a binary representation of whether given states are in given spheres. And possibly extend this idea to hyperspheres in higher dimensions?”

- Randy

Reading Discussion

“How can one determine the most optimal features for a given problem? Is there a systematic approach?”

- Tobias

Reading Discussion

“What I’m curious is, while the book suggests that SGD performs well and that other algorithms may not necessarily offer better performance, I'm curious about how recently developed advanced optimizers like ADAM perform in updating weights in reinforcement learning. Do these newer optimizers still not guarantee improved performance, or is the information in the book possibly outdated?”

- Changgyu

Article Discussion

Reading Discussion

“What are the implications of using human reinforcement for shaping agent behavior in tasks where environmental reward signals are not used?”

- YuanYuan

Reading Discussion

It is said that H can directly define a policy. According to line 7 in algorithm1, I'm wondering what the structure of H is? It seems to me H is consisted of features. But I don't understand how can features define a policy.

- Qing

Reading Discussion

“If a robot trained using the TAMER framework were to become a chef, how would it handle the subjective nature of taste, given that everyone's idea of the "perfect" dish might vary?”

- Jianan

Reading Discussion

“Has there been more follow-up work on whether TAMER is able to solve intractable problems for autonomous agents?”

- Caleb

Reading Discussion

“How might the TAMER framework achieve a balance between human guidance and autonomous decision-making, and how may it solve the ethical issues and difficulties associated with human engagement in the learning process of AI agents?”

- Rafeed

Reading Discussion

“I have some question on how to calculate the credit given by human: it seems like the credit assigned to a state-action pair is the integral of a probability distribution over a time window, but why this is reasonable? Also, how to determine this time window representing the delay? A walk through Algorithm 2 may be helpful.”

- Zichen

Reading Discussion

“How does the TAMER framework handle inconsistent or suboptimal human feedback? Could it be combined with traditional RL techniques to balance human feedback and environmental rewards?”

- Tobias

Reading Discussion

“if an agent is trained using TAMPER framework on one problem, how well can it be transferred to a related problem? Is TAMPER framework problem specific?”

- Zixiao

Reading Discussion

“In the article, it seems like a human would need to provide consistent feedback throughout the training process. Since in some cases training may take many hours, this seems undesirable. Would it make sense to combine the human reinforcement with some other, non-human reliant, reward to remove this constraint?”

- Brennan

Reading Discussion

“Can the TAMER be scaled up to complex real-world scenarios where multiple agents or systems are interacting simultaneously?”

- Mingjia

Reading Discussion

“Can TAMER or similar shaping approaches be used early on to create beneficial initial conditions for the policy, to then be improved over many more iterations by something like policy iteration or a genetic algorithm?”

- Grayson

Reading Discussion

“I found it quite surprising the effectiveness of human feedback because I have always imagined training to be faster because it can be calculated much faster but with human feedback that would slow things down significantly. I am also curious the exact user experience that is required to give human feedback?”

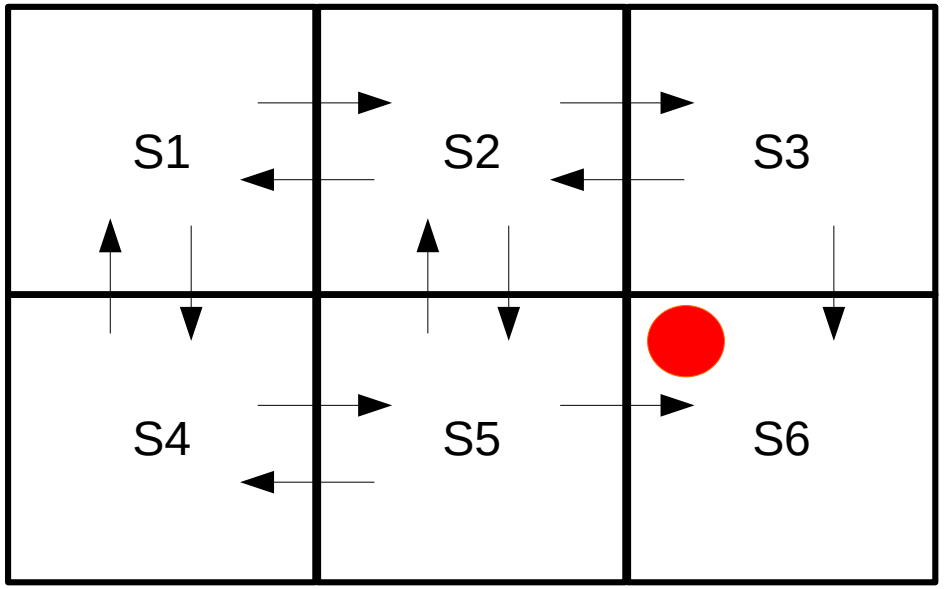
- Daniel

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

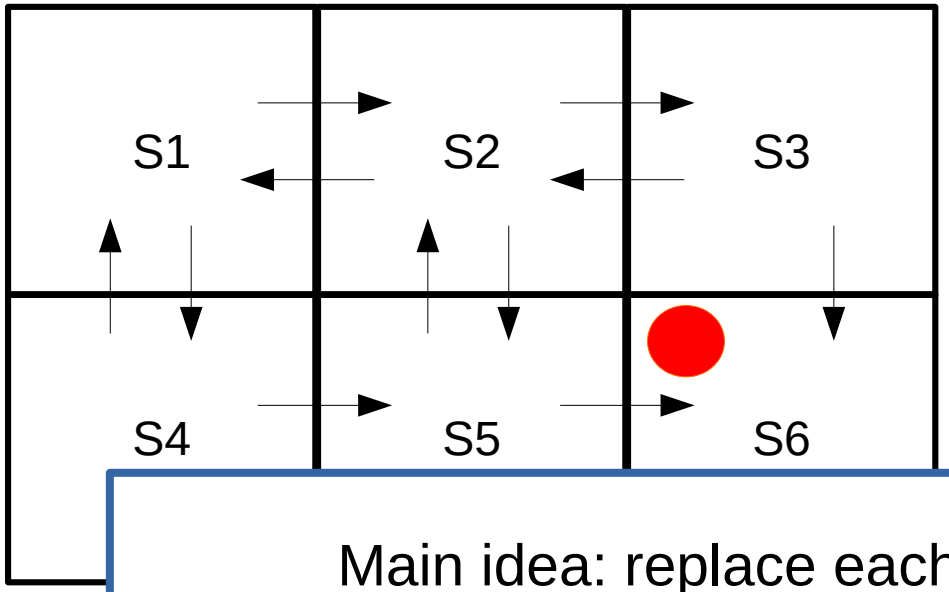
Update rule upon executing action a, ending up in state s' and observing reward r :

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

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

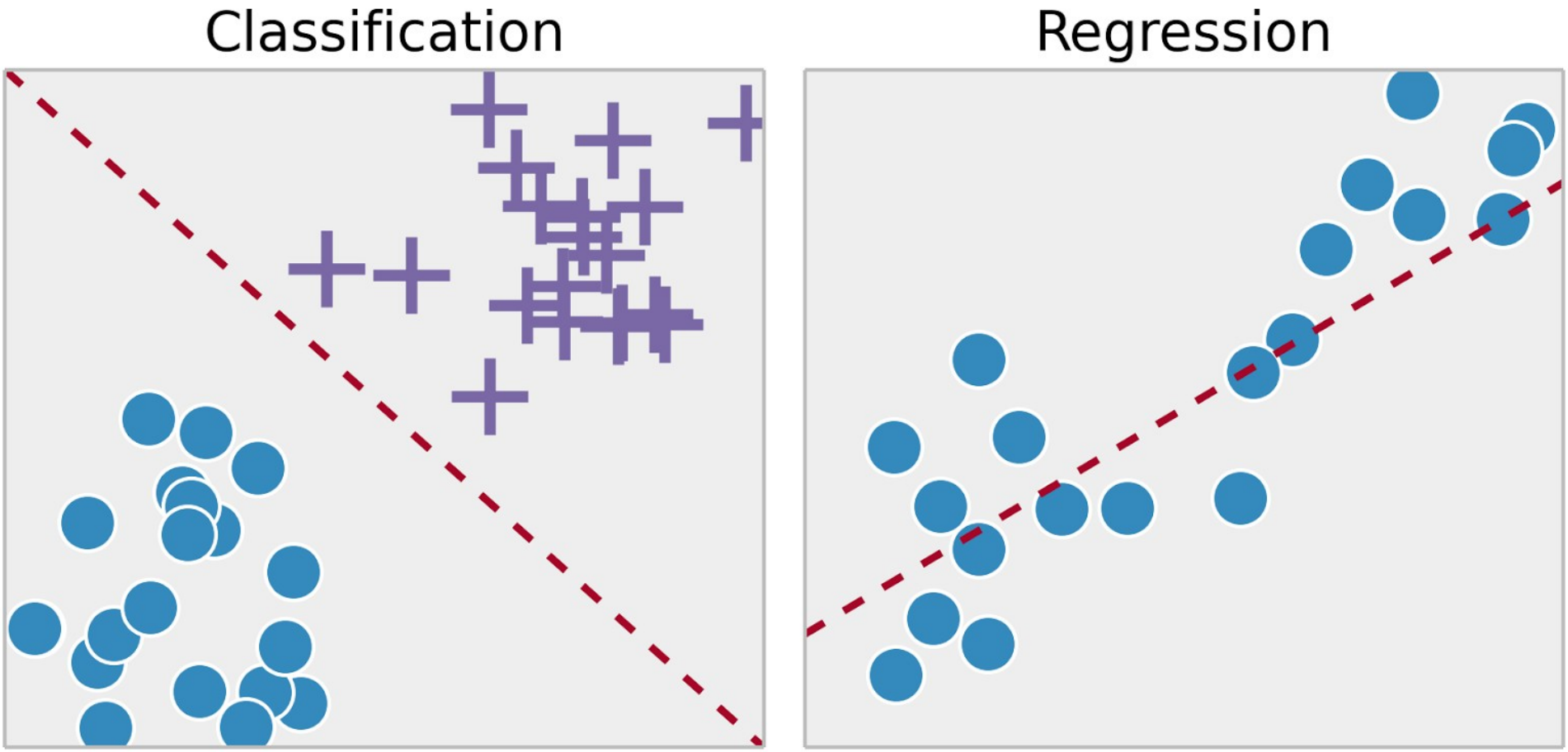
S1	right	25
S1	down	25
S2	right	50
S2	left	12.5
S2	down	50
S3	left	25
0.5, -0.7, 0.2, …, 0.9		100
S4	up	12.5
S4	right	50
S5	left	25
S5	up	25
S5	right	100

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)

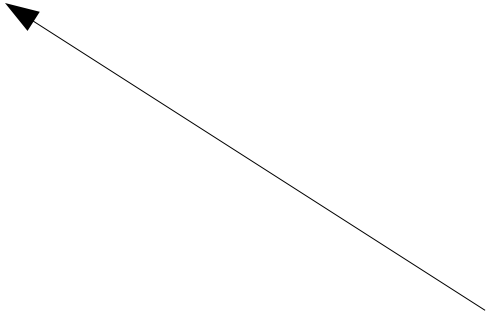
Connection to Supervised ML



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

Linear Q-Function Approximation

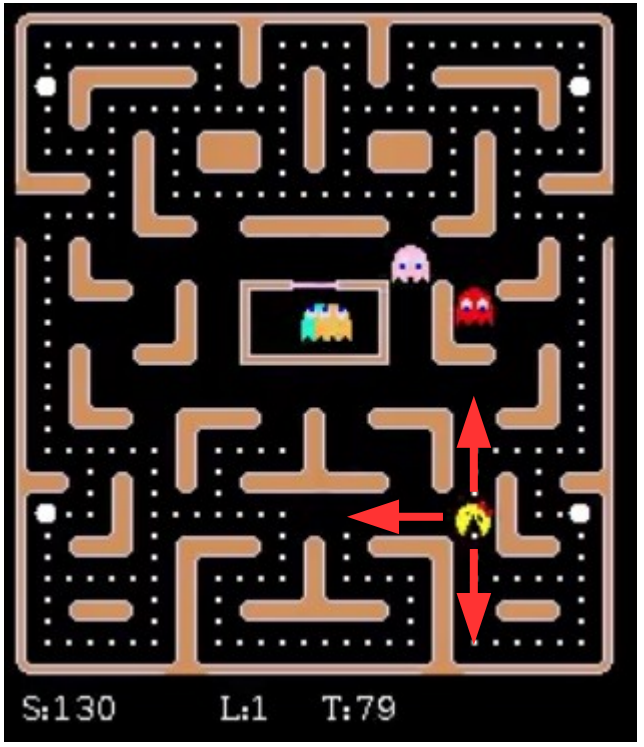
$$Q^*(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s'} \mathcal{P}(s'|s,a) \max_{a'} Q^*(s',a')$$



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

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

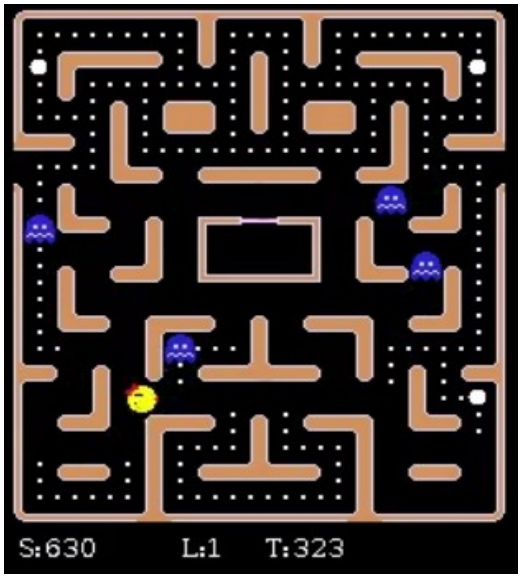
Example: Ms. Pac-man

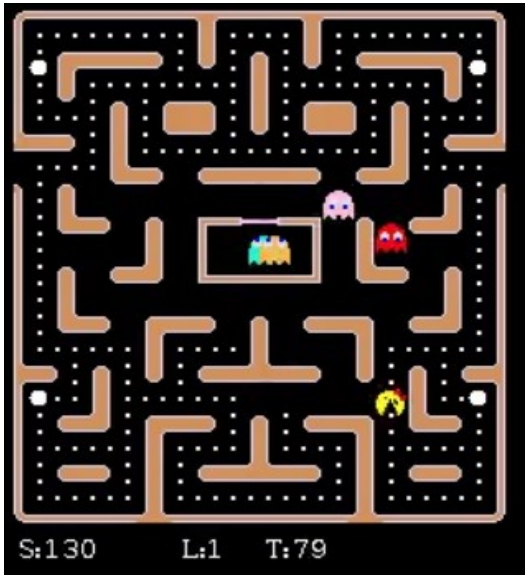


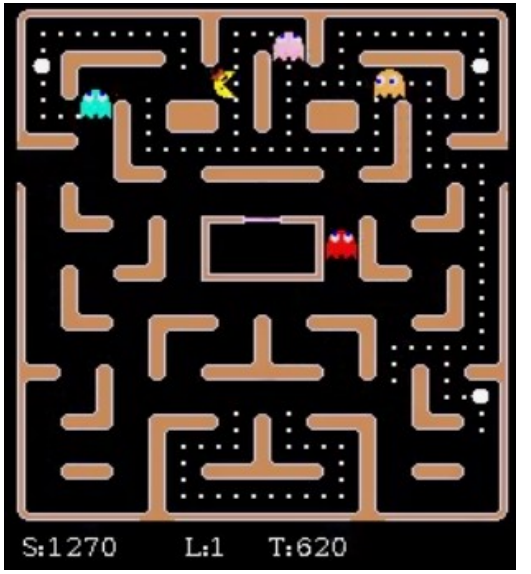
The problem: for a given action and the current configuration, compute a fixed-length feature vector

Each feature must have some semantic “meaning”

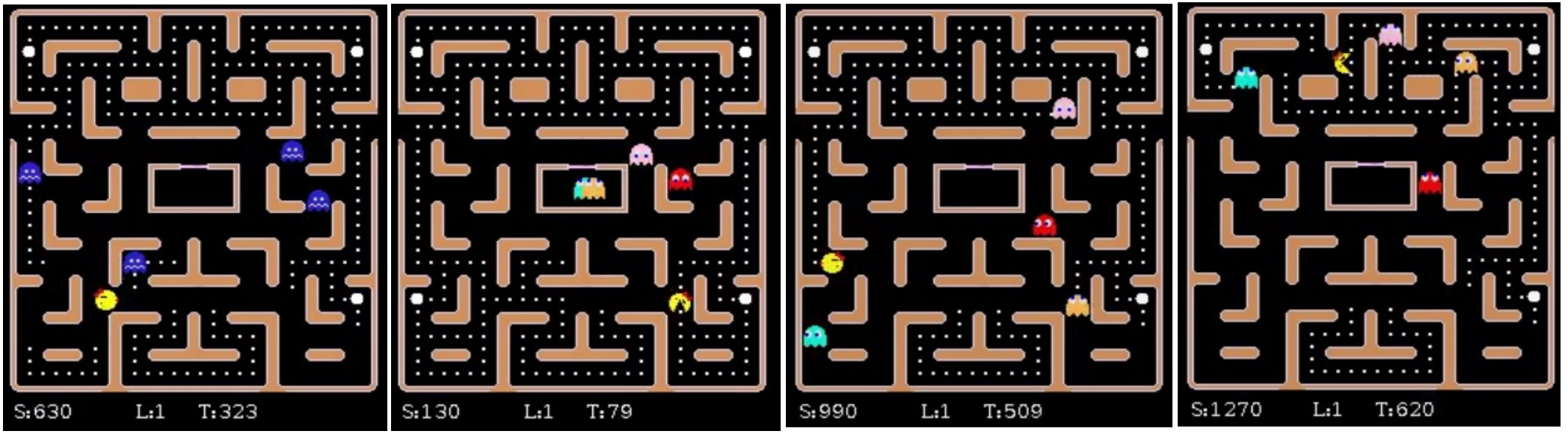
Example Configurations







Small group activity: feature engineering



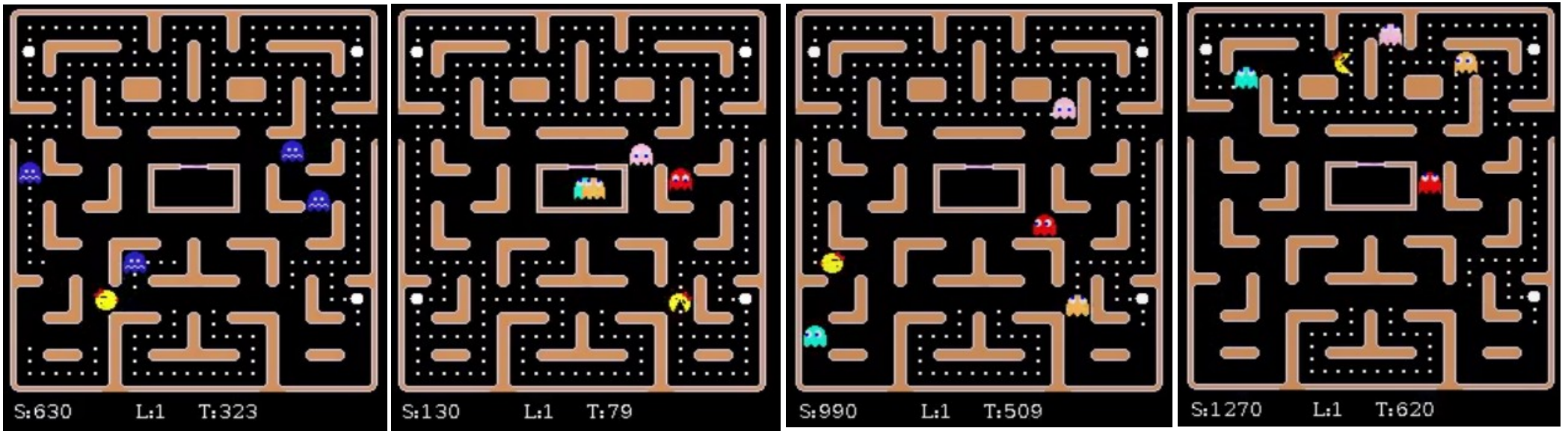
Be the feature engineer: given a configuration and a cardinal direction, design the feature types that describe how the world “looks like” in that direction; assume you have access to the underlying game simulator; the board itself is a graph with nodes and edges and for each node, you know whether there 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