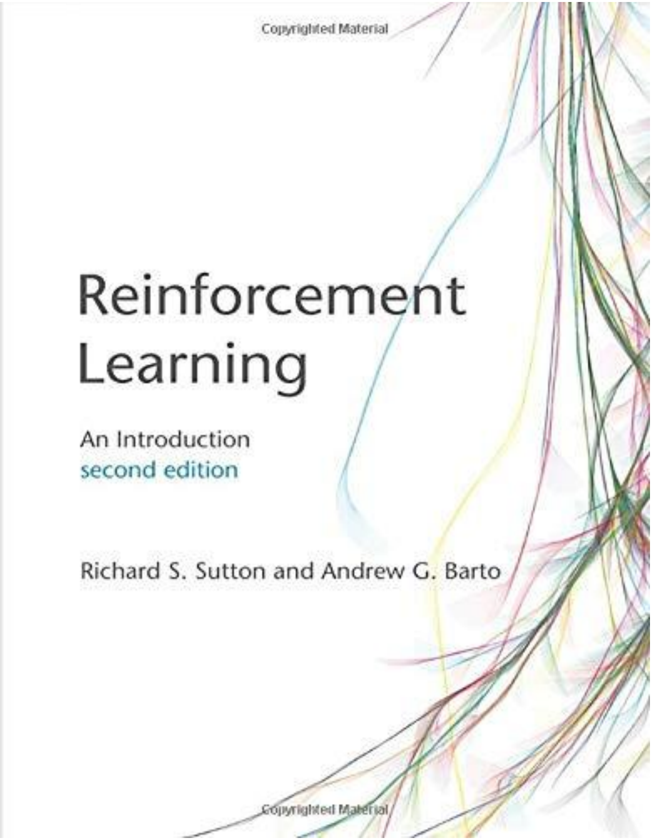
COMP 3200
Artificial Intelligence

MEMORIAL UNIVERSITY

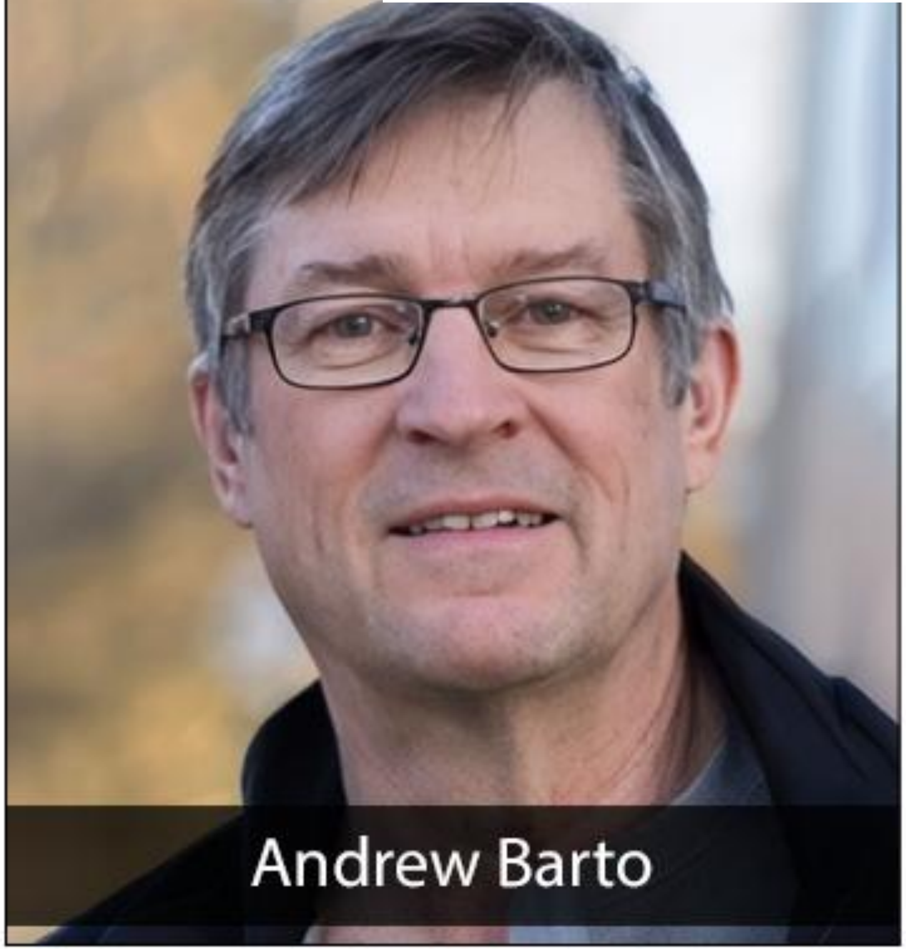
Lecture 16
Intro to Reinforcement Learning

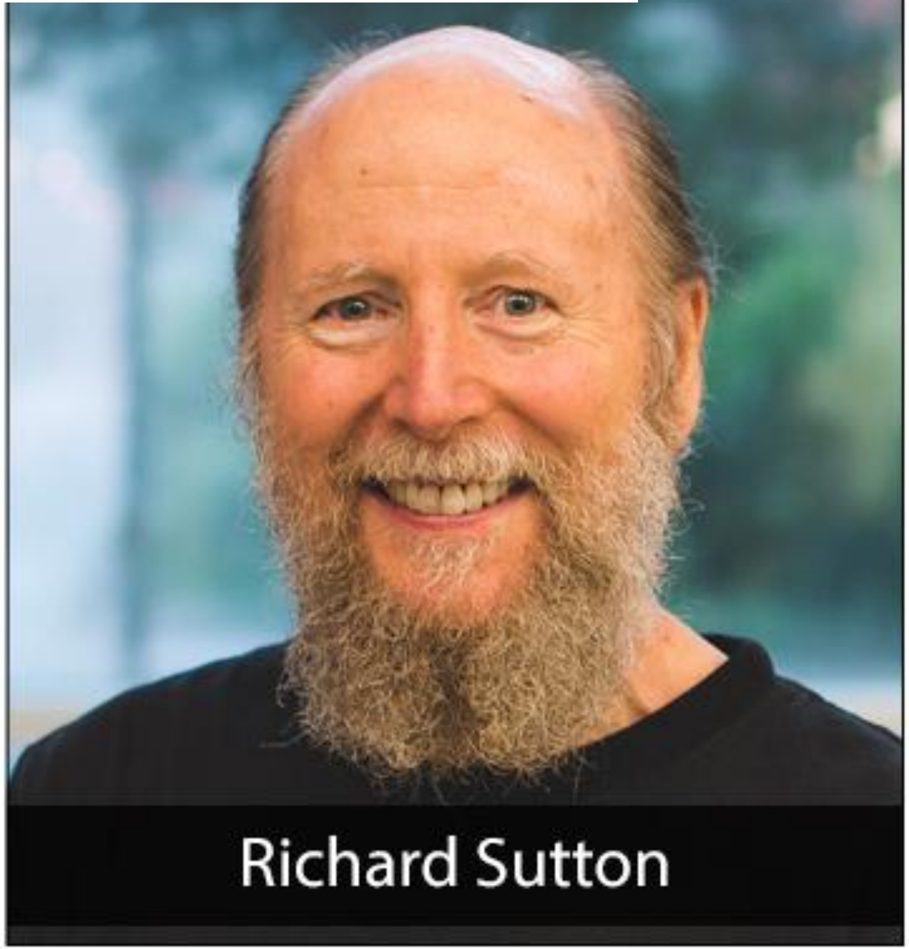
Incredible Textbook

- Sutton & Barto
- Best textbook ever
- Very understandable, intuitive explanations
- Highly recommend reading and following along
- http://incompleteideas.net/book/the-book.html



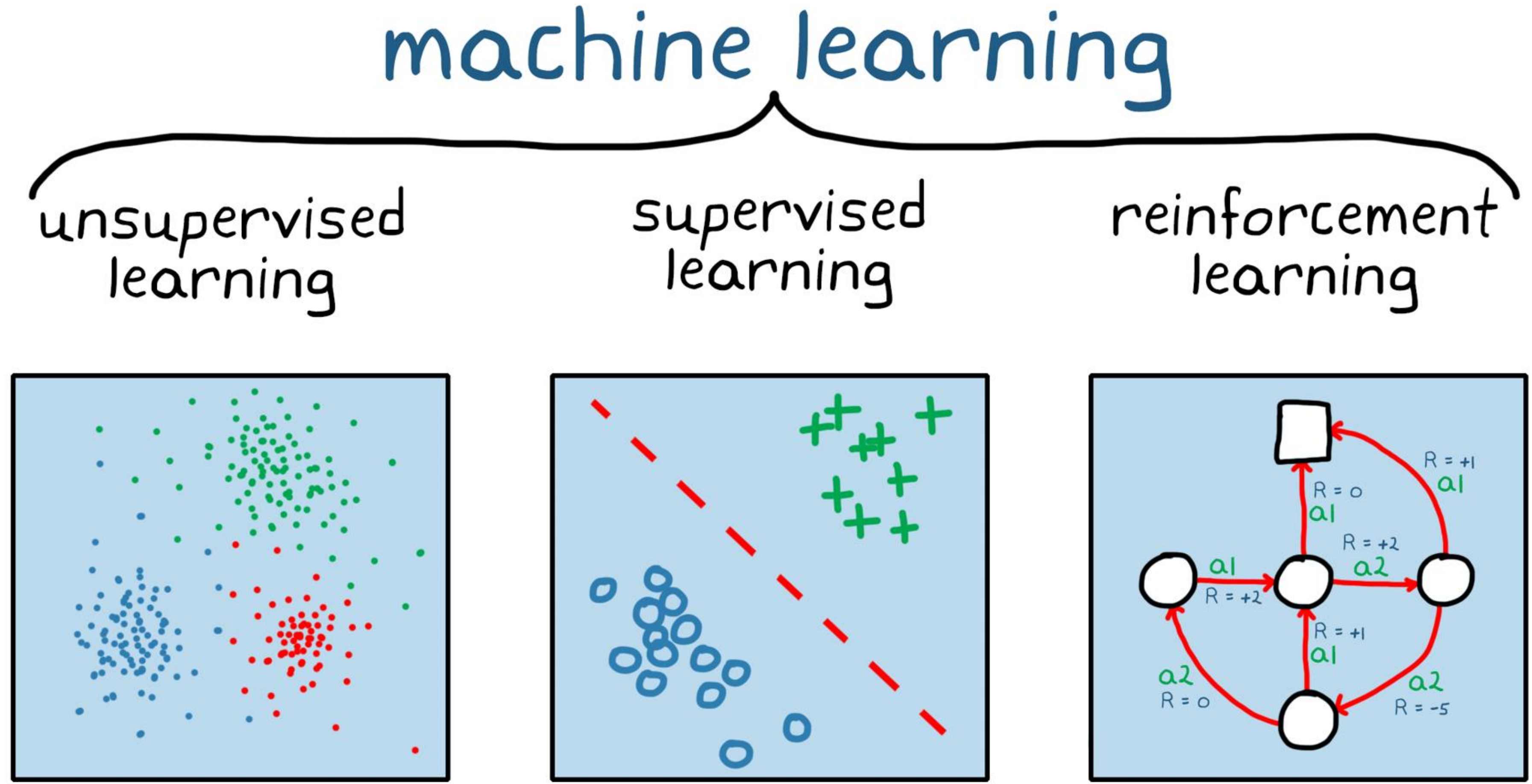
ACM A.M. Turing Award (2024)





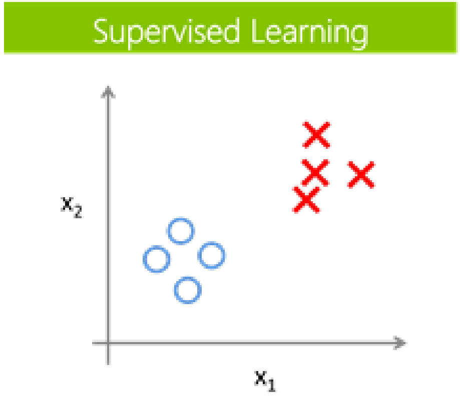
Reinforcement Learning

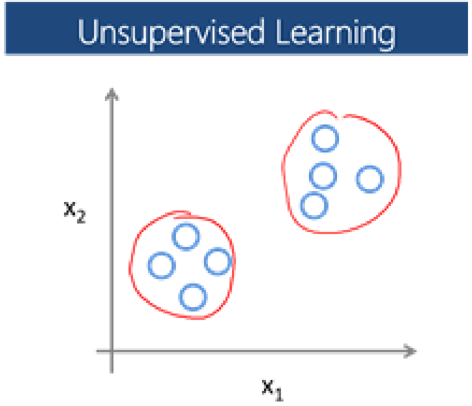
- RL is a sub-area of Machine Learning
- Inspired by natural systems / psychology
- Learn via interaction with environment
- Agents receive rewards
  - Good actions = positive reward
  - Bad actions = negative reward
- RL agent goal is to maximize rewards


machine learning
unsupervised learning
supervised learning
reinforcement learning
R = +1
a1
R = 0
a1
R = +2
a2
a1
R = +2
R = +1
a1
a2
R = 0
a2
R = -5

Machine Learning

- Supervised Learning
  - Learn from training data
  - Given inputs / true outputs
  - Learn to predicts outputs for new unseen input samples
- Unsupervised Learning
  - Let the computer learn from data without labeled outputs





Supervised Learning in ML

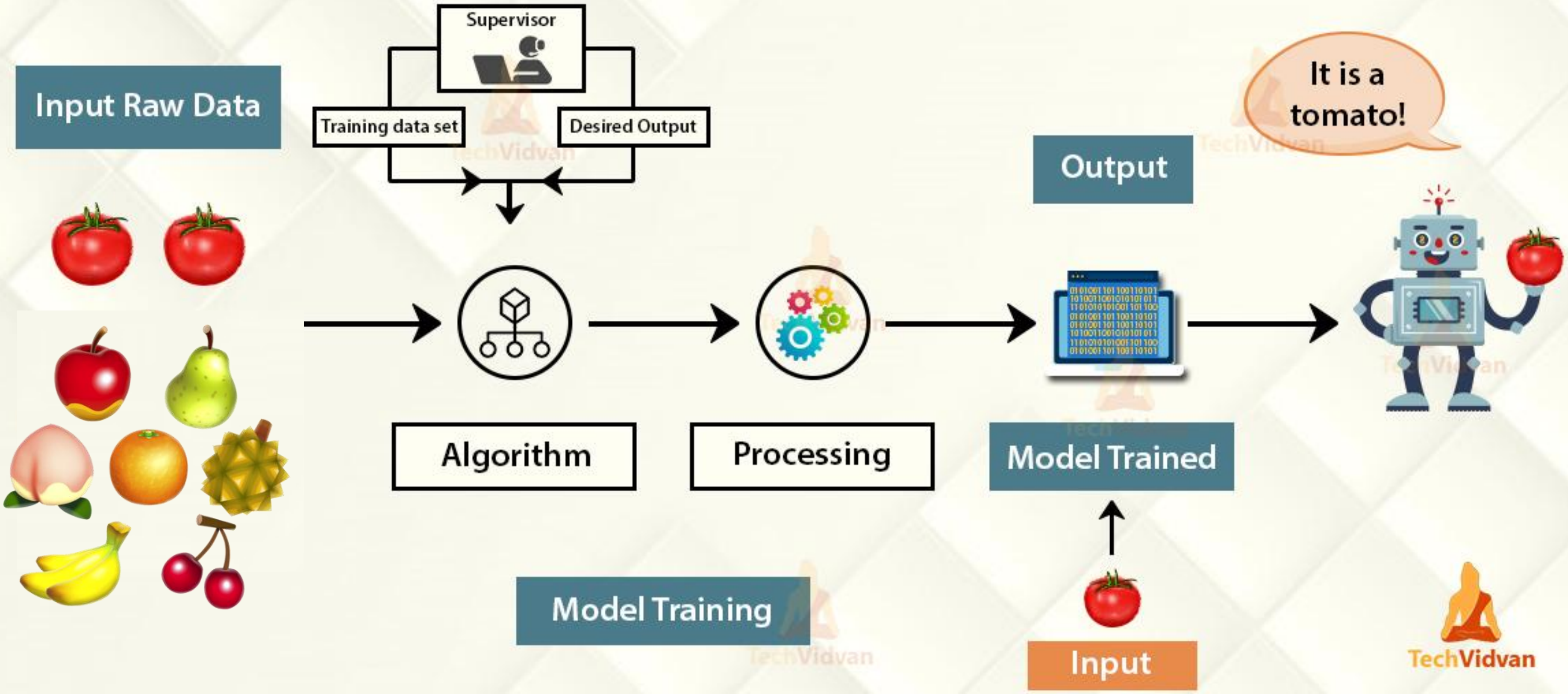
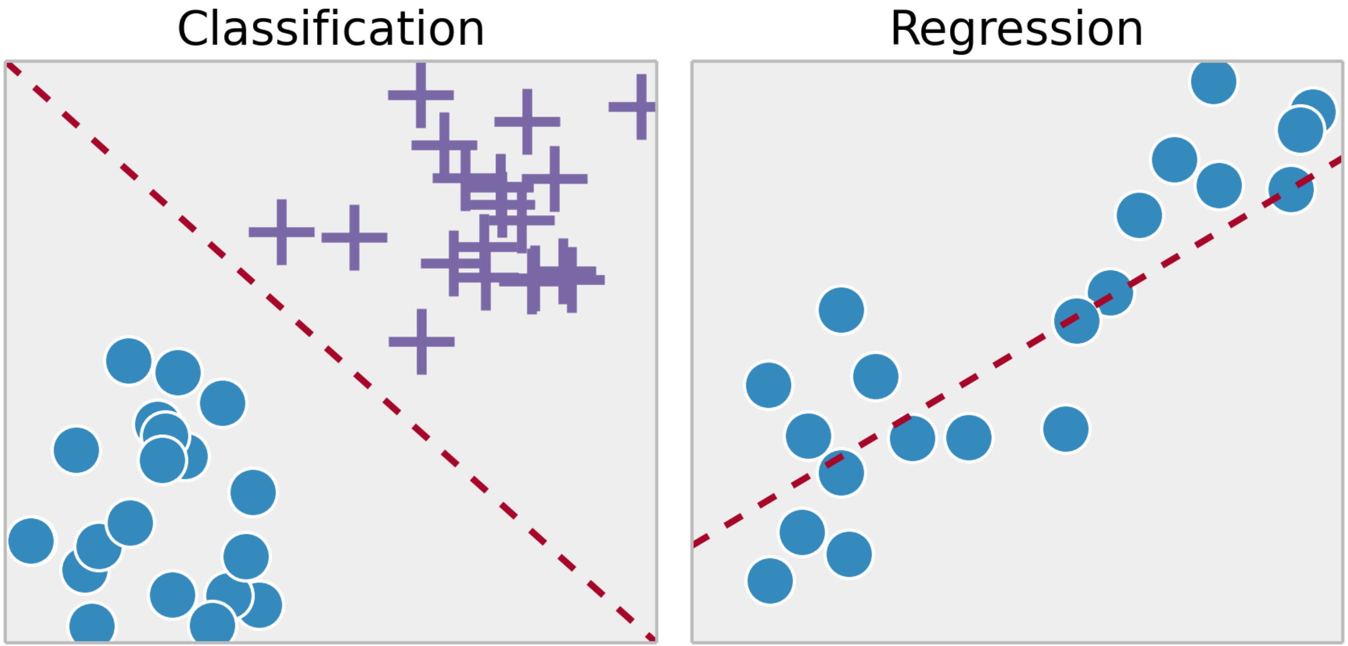
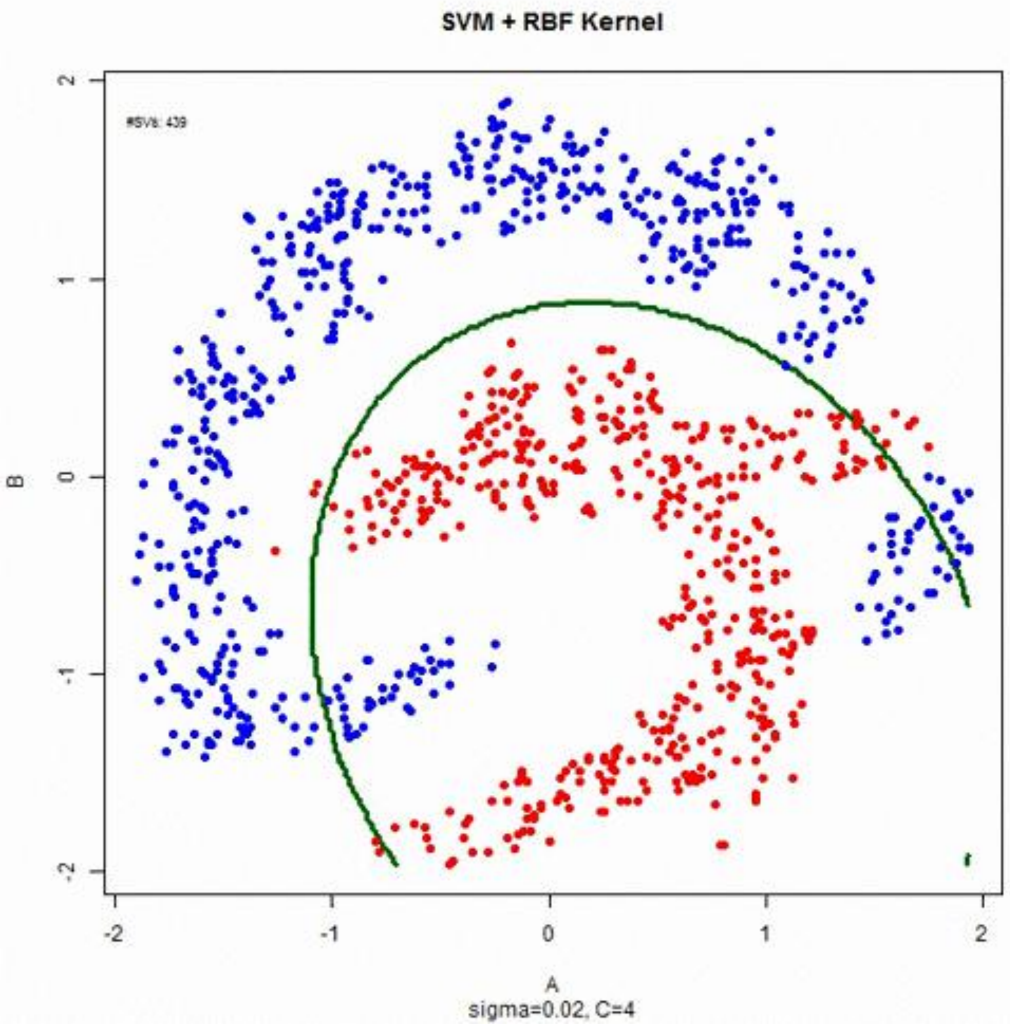
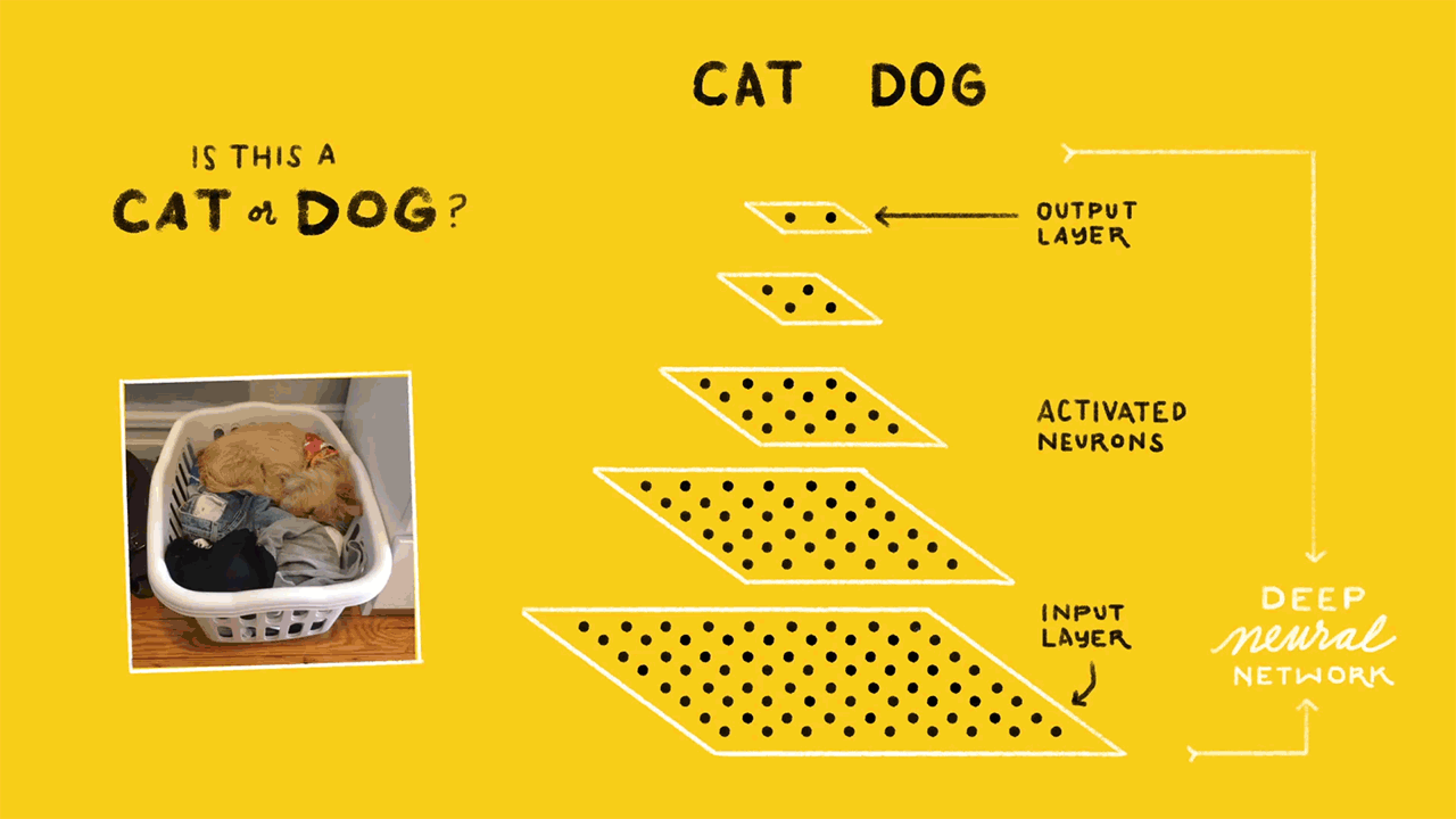


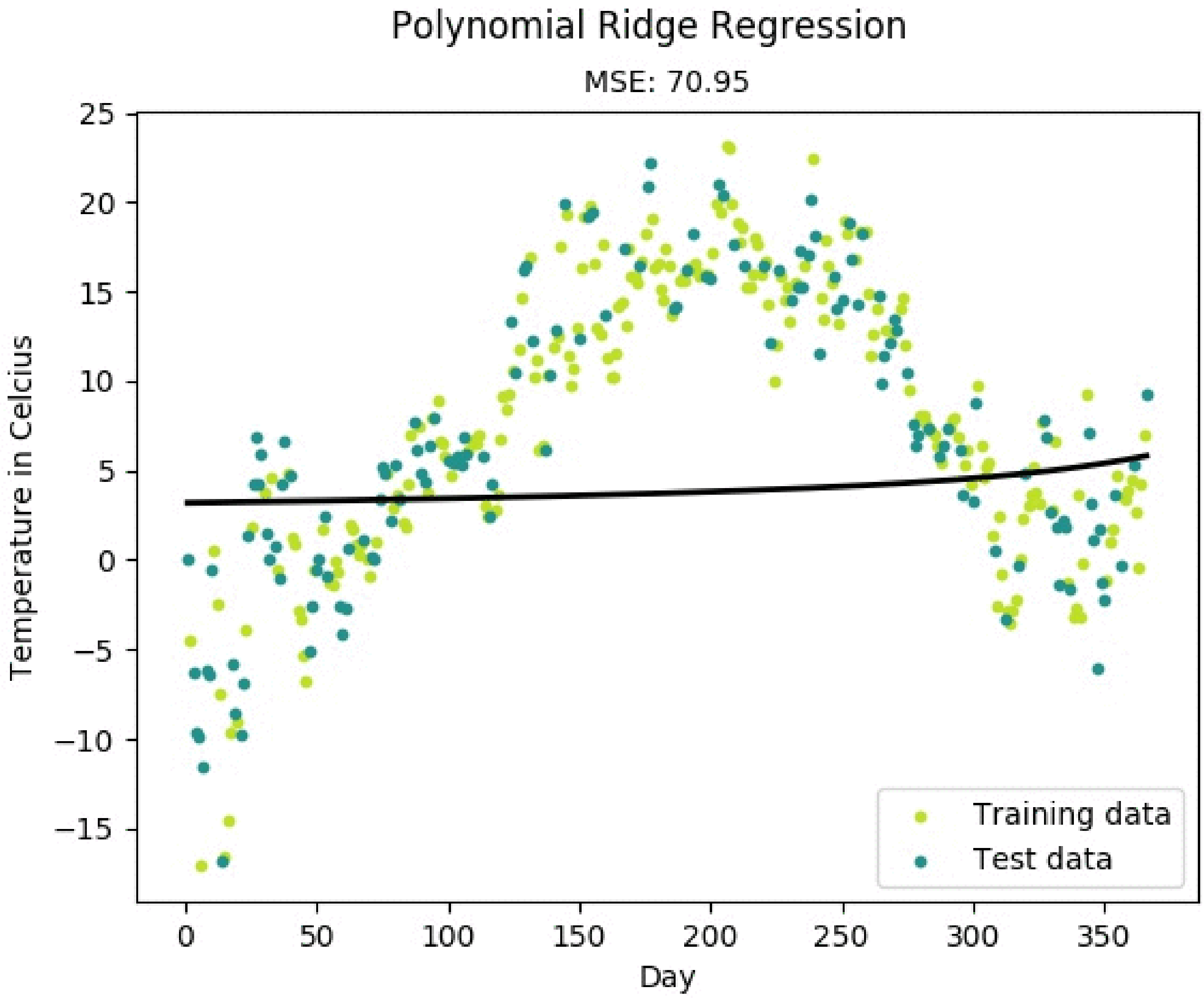
Image src: https://techvidvan.com/tutorials/supervised-learning/

Supervised Learning Examples

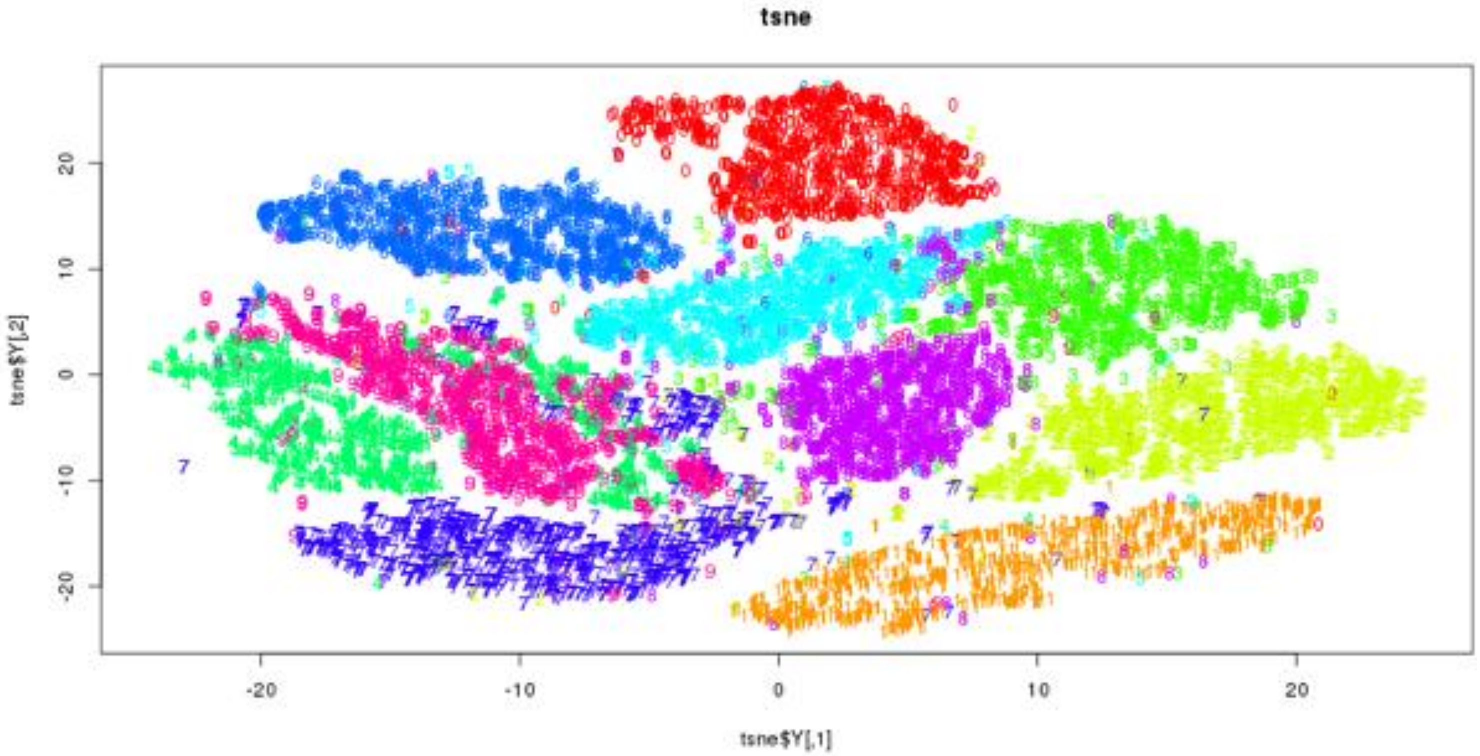


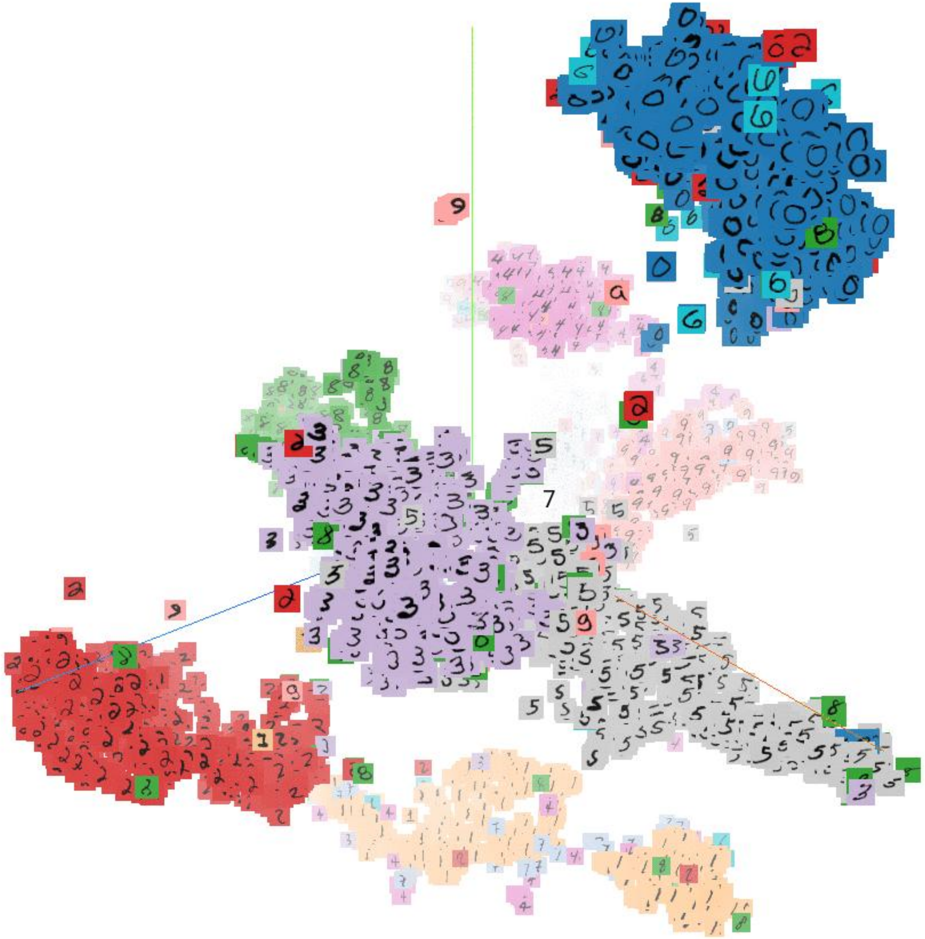

SVM + RBF Kernel
#SVs: 439
2
1
0
-1
-2
B
-2
-1
0
1
2
A
sigma=0.02, C=4


IS THIS A
CAT or DOG?
CAT DOG
OUTPUT LAYER
ACTIVATED NEURONS
INPUT LAYER
DEEP neural NETWORK


Polynomial Ridge Regression
MSE: 70.95
Temperature in Celcius
Day
25
20
15
10
5
0
−5
−10
−15
0
50
100
150
200
250
300
350
Training data
Test data

Unsupervised Learning Example




7

Reinforcement Learning

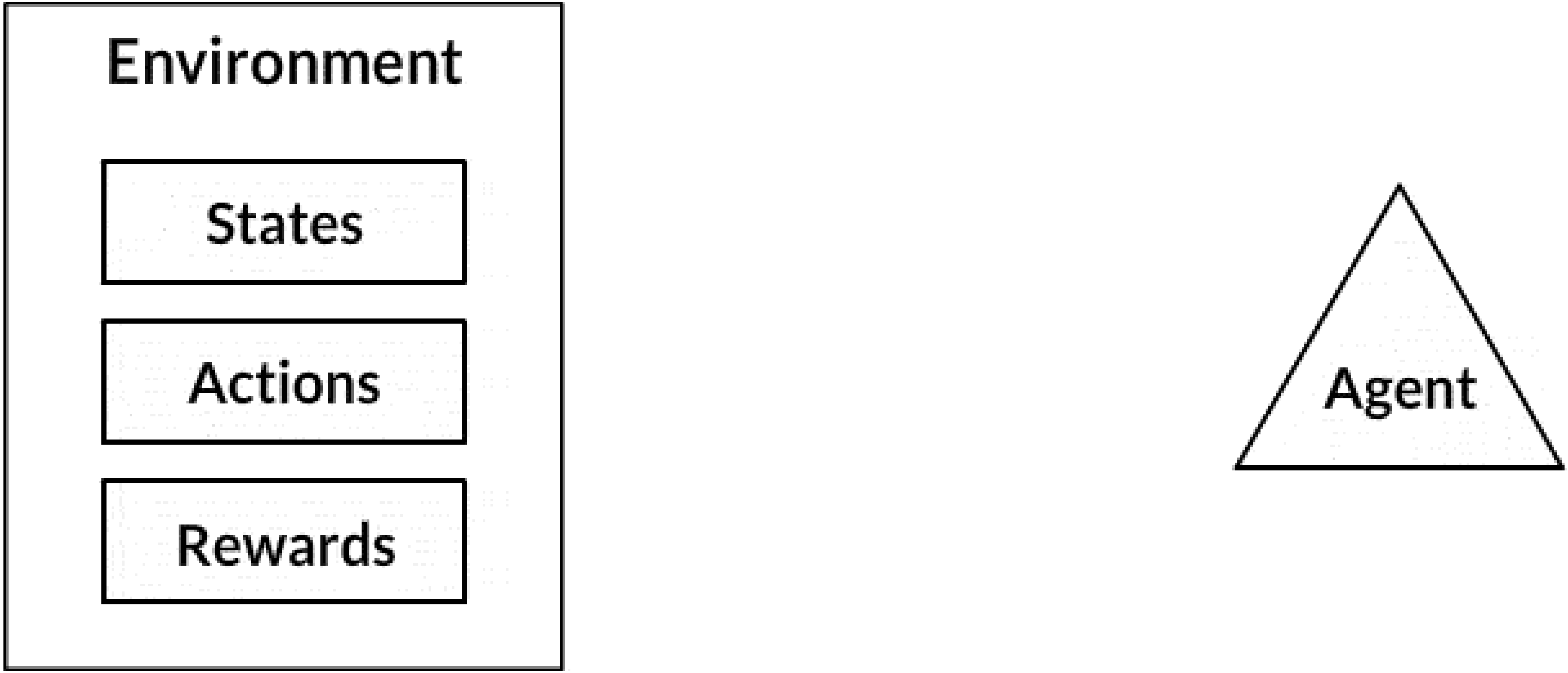
- RL is not an *algorithm*
- RL is more of a *problem specification*

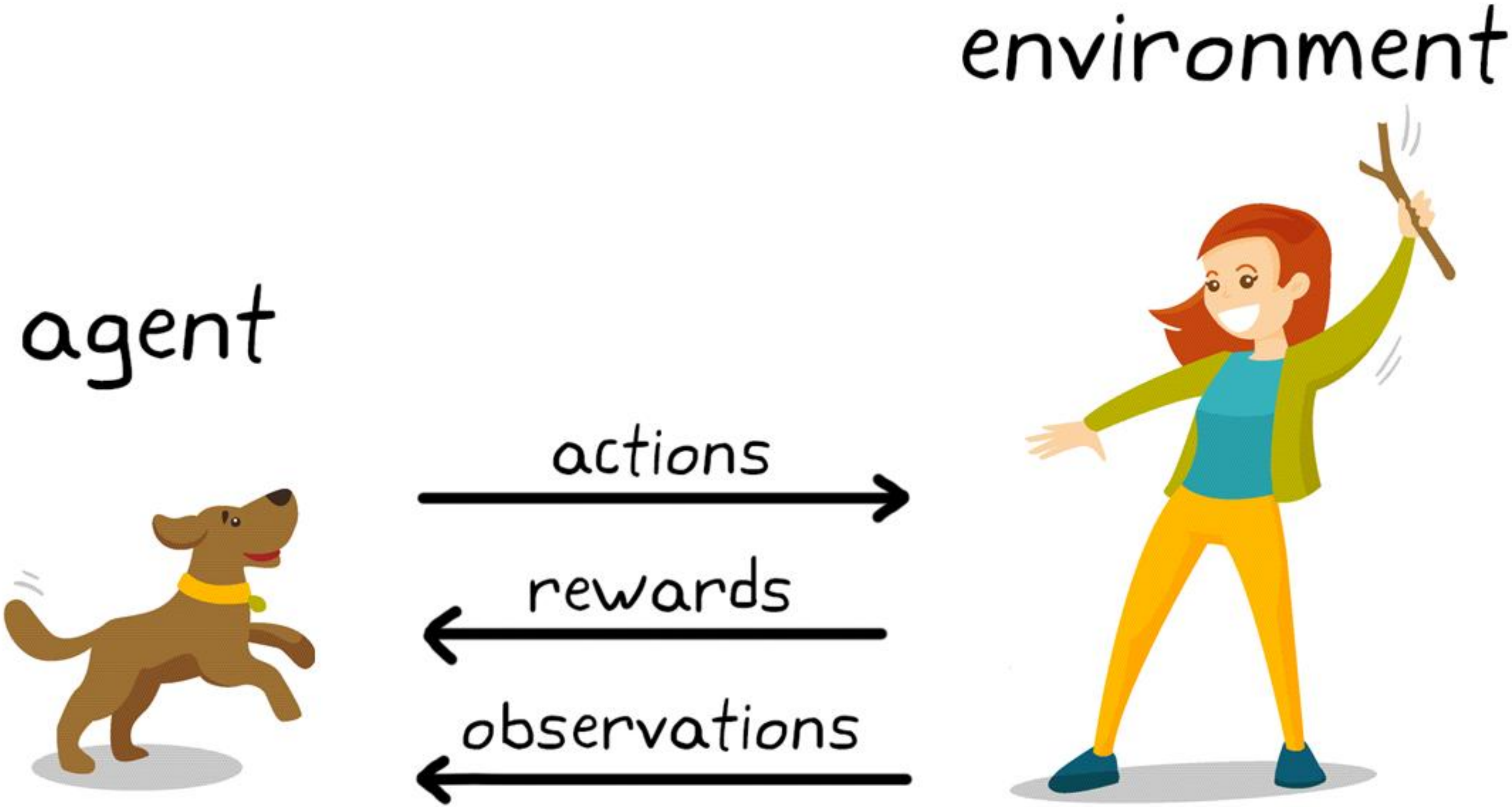
- Many algorithmic techniques can be used to solve Reinforcement Learning problems

- "RL Method" – solves a RL problem

Reinforcement Learning

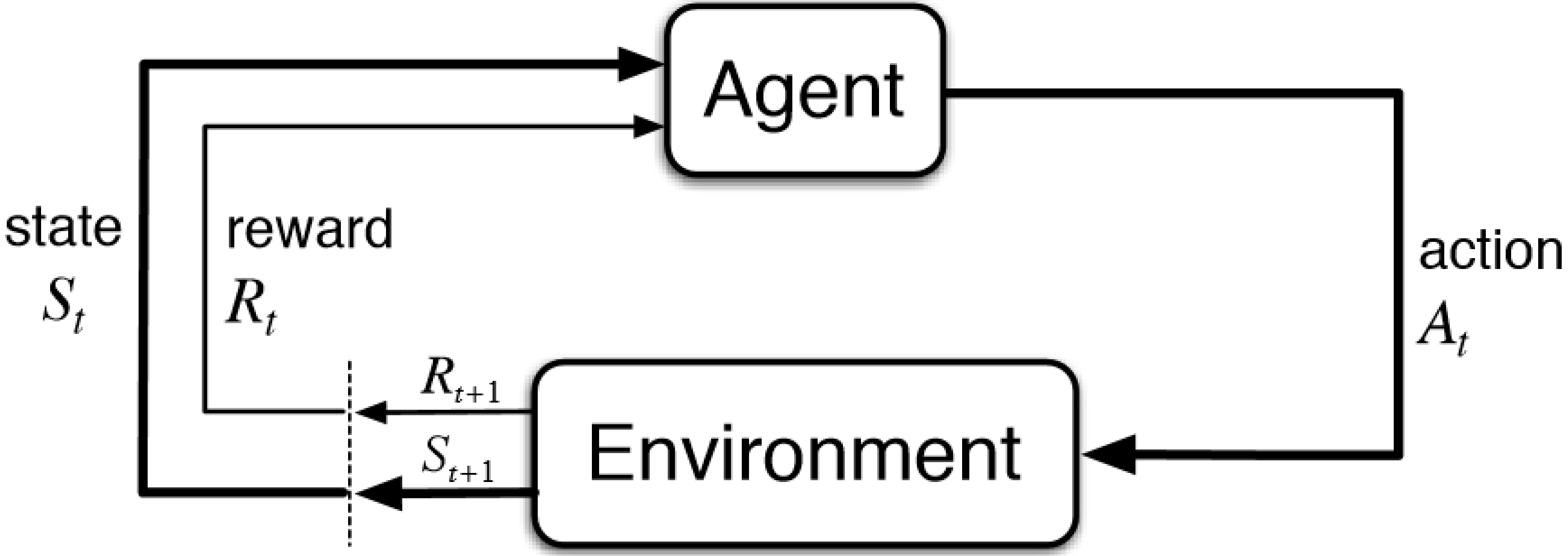
- Learning via interaction with environment
- No explicit ‘teacher’ for learning
- Agent take some actions, and receives a reward signal from the environment
- Learning via interaction teaches us about cause and effect, consequences of actions, what to do to achieve goals


Environment
States
Actions
Rewards
Agent



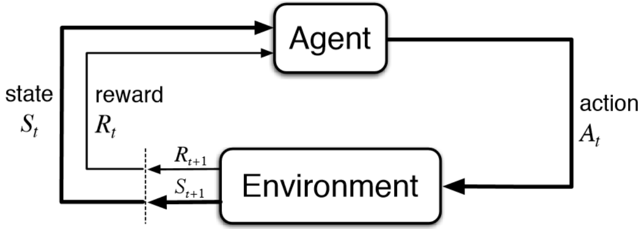
Image(s) - https://www.mathworks.com/discovery/reinforcement-learning.html

Reinforcement Learning

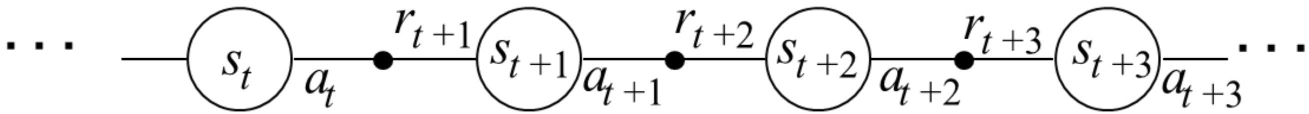


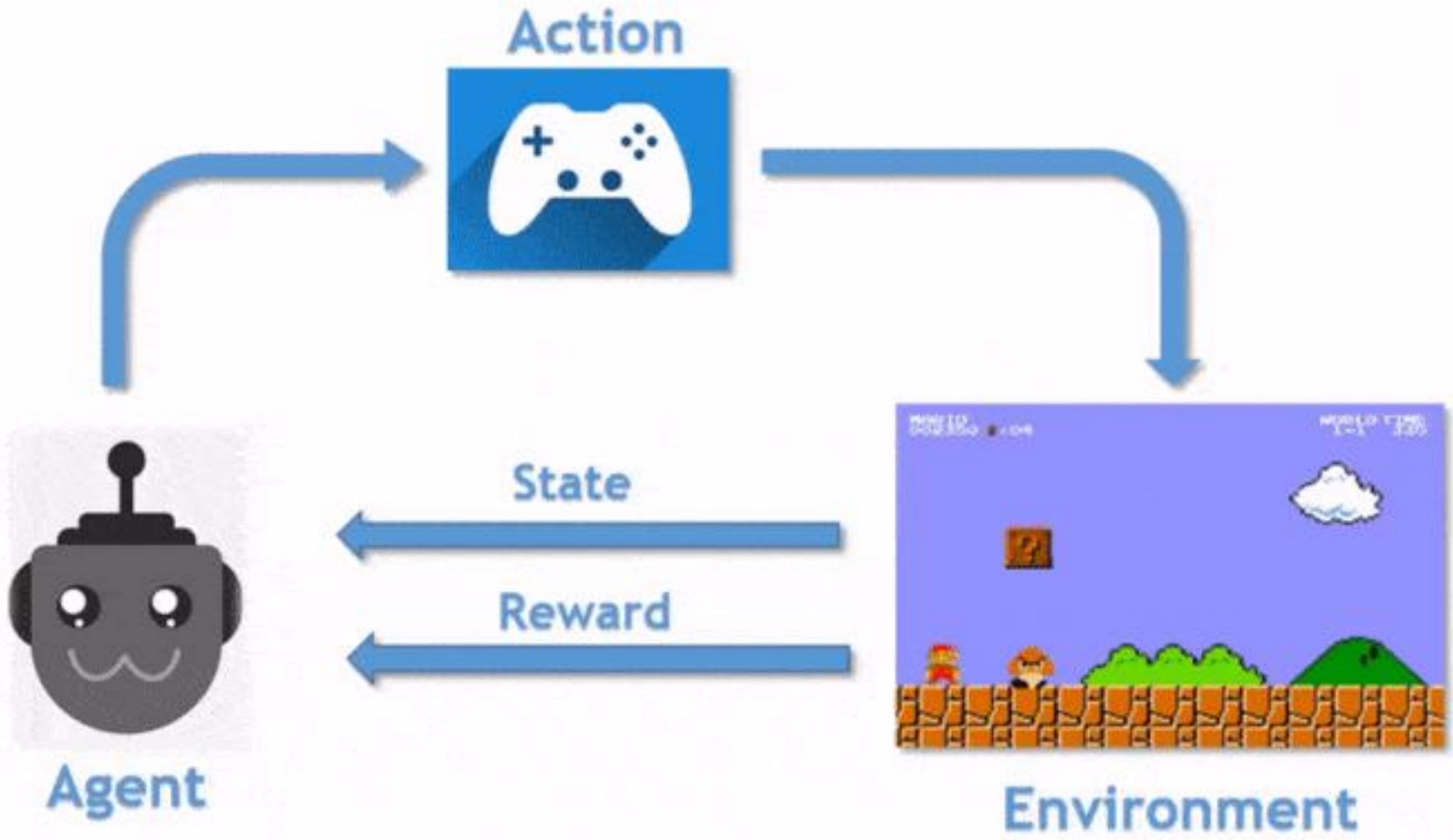
Reinforcement Learning

- Agent / Environment interact in discrete time steps t=0,1..n

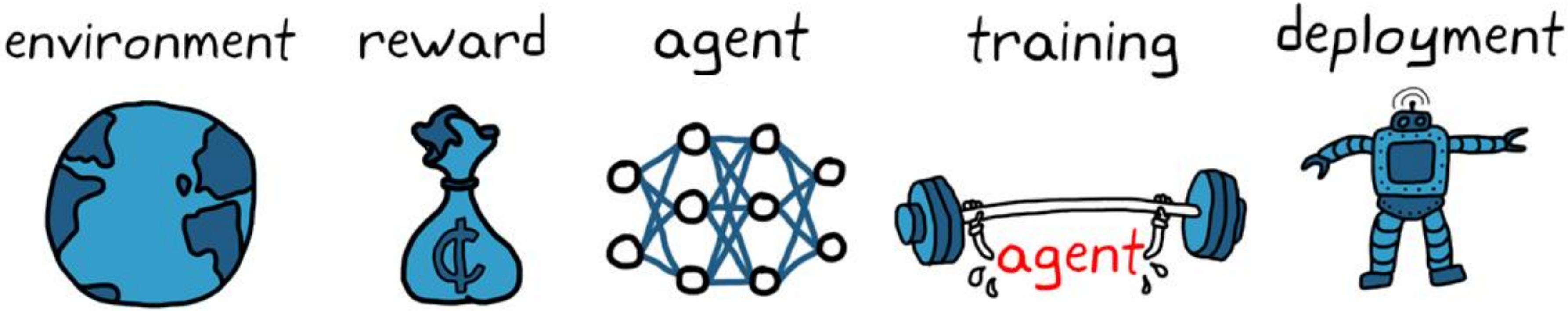


- Agent observes a state at step t — $s_t \in S$
- Agent produces action at step t — $a_t \in A(s_t)$
- Environment gives resulting reward — $r_{t+1} \in R$
- Transition to next state — $s_{t+1} \in S$




Action
State
Reward
Agent
Environment

RL Workflow

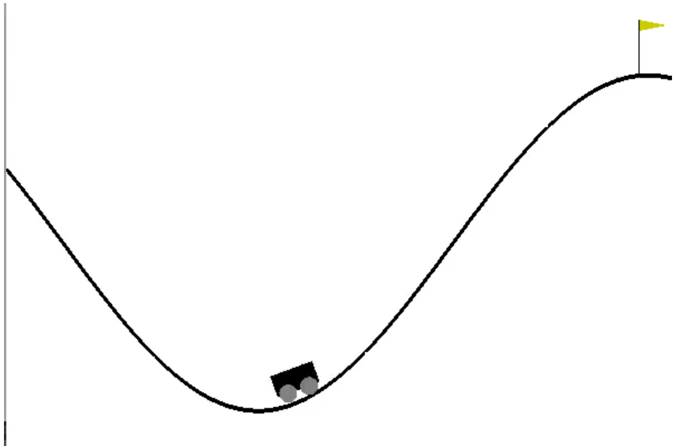


RL Example 1

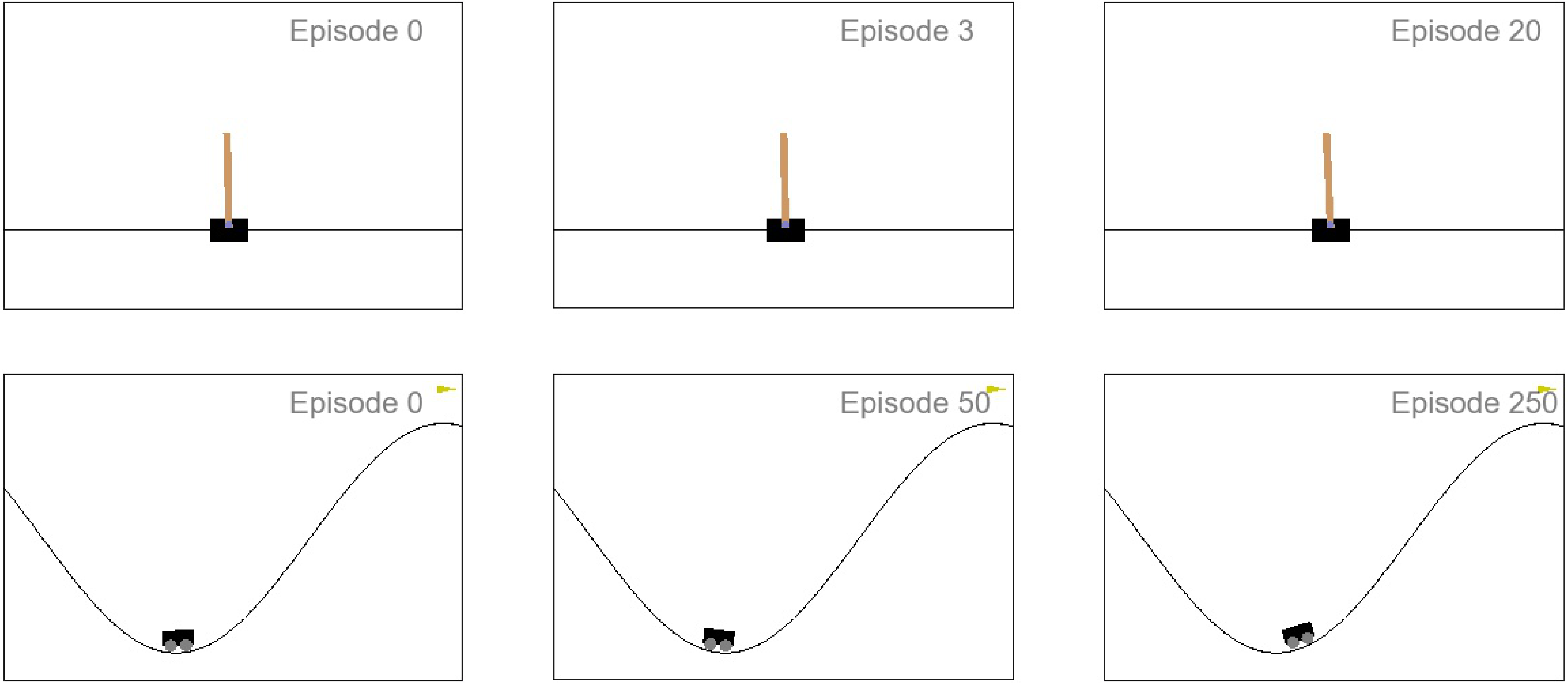
- Cart-Pole Problem
- Balance the pole on the moving base as long as possible

- Actions: move left, move right
- Rewards: +1 for each time step balanced
  - Maximize Reward = Maximize Balance Time
- Episodic task: terminal state when pole falls

RL Example 2



- Mountain Car Problem
- Escape to the top of the mountain as fast as possible
- Actions: accel left, accel right
- Rewards: -1 for each time step
  - Maximize Reward = Minimize time to goal
- Episodic task: terminal state when reach goal


Episode 0
Episode 3
Episode 20
Episode 0
Episode 50
Episode 250

Elements of RL Problems

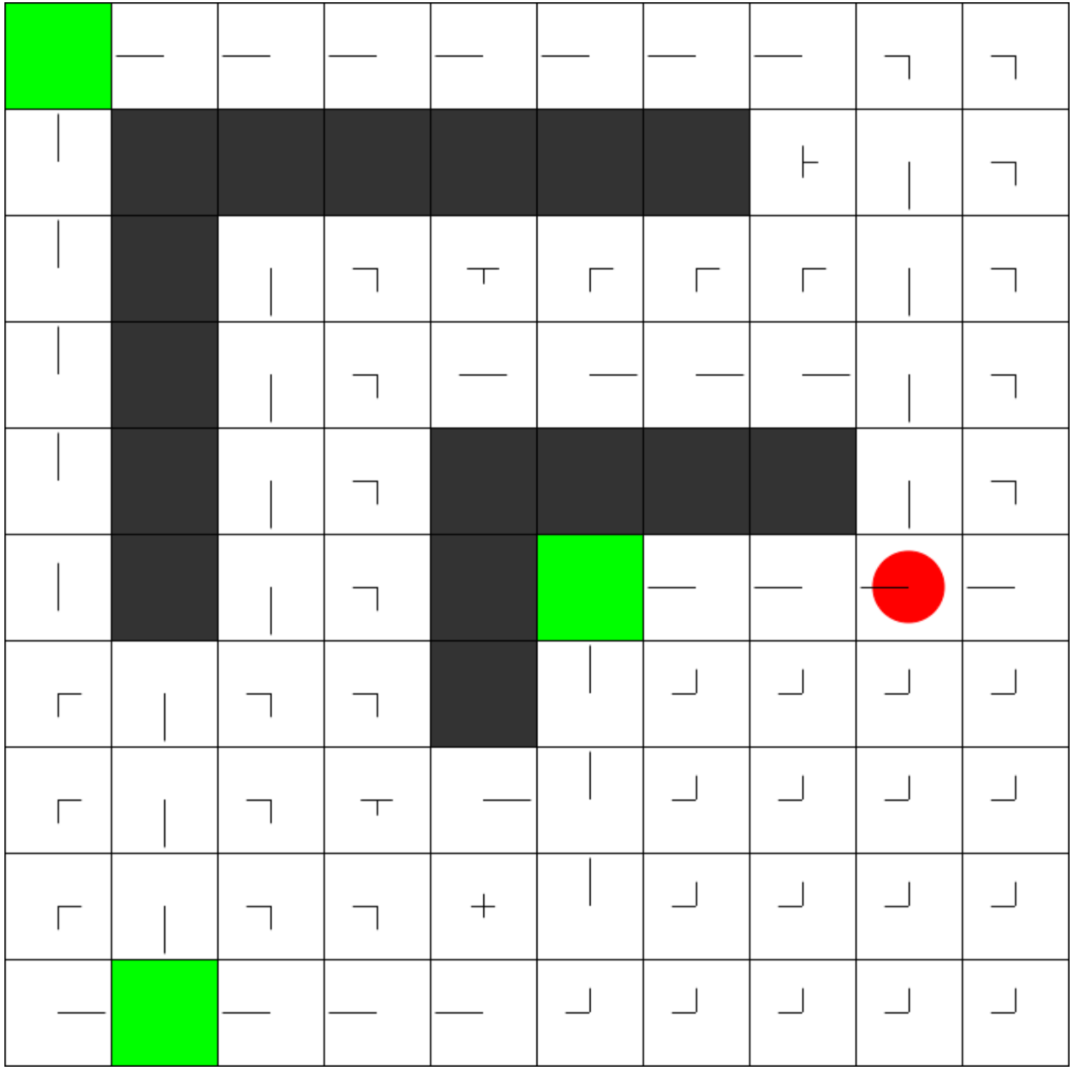
- Agent
- Environment
- Reward Function
- Policy
- Value Function
- Model of Environment (optional)

Elements of RL: Policy

- A map from states of the environment to actions to be taken at those states
- Map be a simple look-up table, or a complex computation such as neural net
- The core of an RL agent, defines behavior
- May be deterministic or stochastic
  - Move left with 50% of the time, up 50%

Elements of RL: Policy

- Policy at time step t = π_t
- Policy maps from states to the probability of taking an action at that state
- $\pi_t(s,a)$ = probability that agent takes action $a_t = a$ when state $s_t = s$
- RL methods specify how an agent changes its policy over time as it learns
- Agent's goal is to form a policy that gets as much reward as it can over time



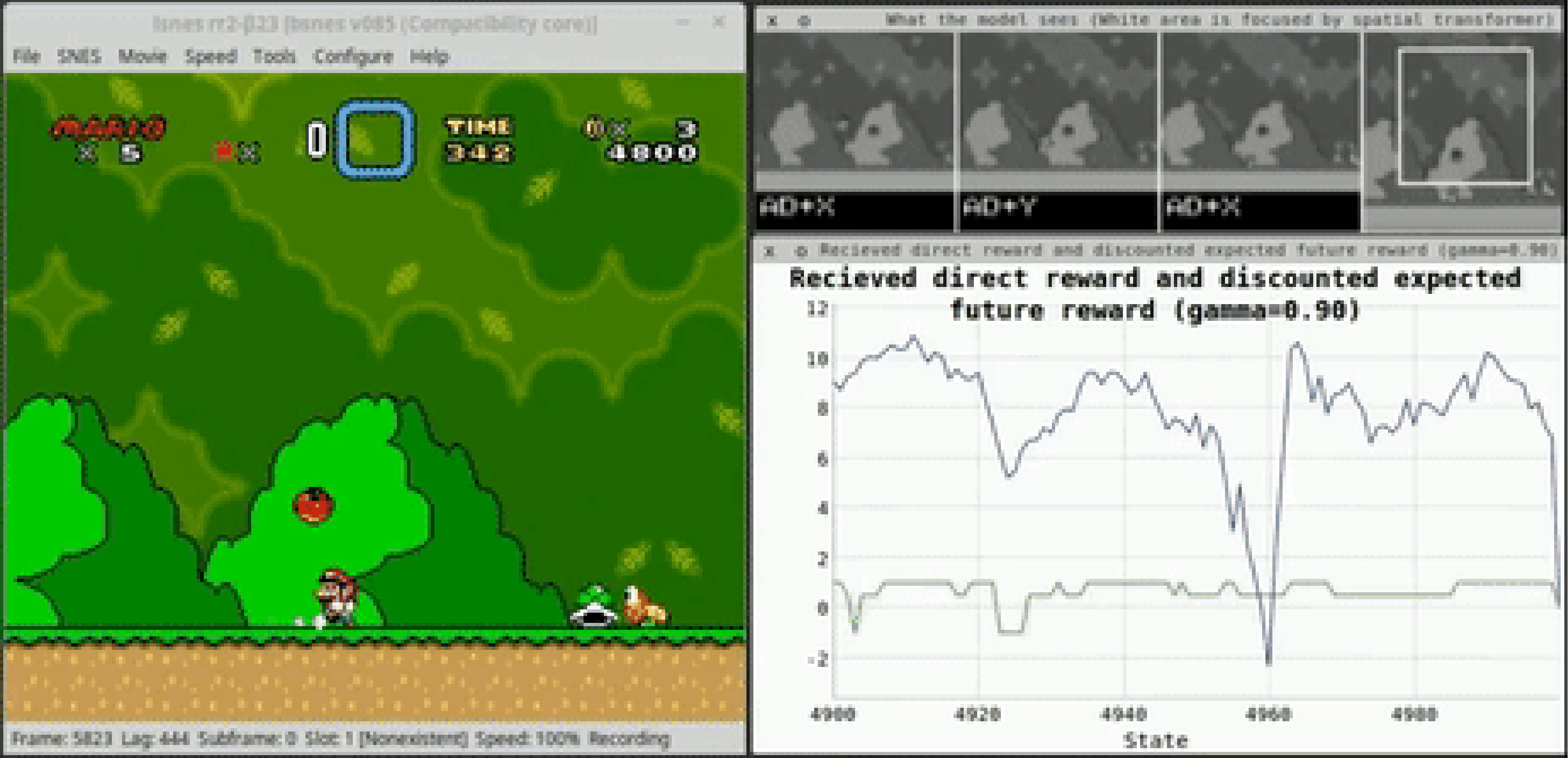
	ADVANCED BLACKJACK STRATEGY TABLE									
	Dealer's First Card									
Your Hand	2	3	4	5	6	7	8	9	10	A
18+	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND
17	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND
16	STAND	STAND	STAND	STAND	STAND	HIT	HIT	HIT	HIT	HIT
15	STAND	STAND	STAND	STAND	STAND	HIT	HIT	HIT	HIT	HIT
14	STAND	STAND	STAND	STAND	STAND	HIT	HIT	HIT	HIT	HIT
13	STAND	STAND	STAND	STAND	STAND	HIT	HIT	HIT	HIT	HIT
12	HIT	HIT	STAND	STAND	STAND	HIT	HIT	HIT	HIT	HIT
11	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	HIT
10	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	HIT	HIT
9	HIT	DOUBLE	DOUBLE	DOUBLE	DOUBLE	HIT	HIT	HIT	HIT	HIT
8	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT
7	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT
6	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT
5	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT
Soft 20	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND
Soft 19	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND
Soft 18	STAND	DOUBLE	DOUBLE	DOUBLE	DOUBLE	STAND	STAND	HIT	HIT	HIT
Soft 17	HIT	DOUBLE	DOUBLE	DOUBLE	DOUBLE	HIT	HIT	HIT	HIT	HIT
Soft 16	HIT	HIT	DOUBLE	DOUBLE	DOUBLE	HIT	HIT	HIT	HIT	HIT
Soft 15	HIT	HIT	DOUBLE	DOUBLE	DOUBLE	HIT	HIT	HIT	HIT	HIT
Soft 14	HIT	HIT	HIT	DOUBLE	DOUBLE	HIT	HIT	HIT	HIT	HIT
Soft 13	HIT	HIT	HIT	DOUBLE	DOUBLE	HIT	HIT	HIT	HIT	HIT
Pair A	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT
Pair 10	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND
Pair 9	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	STAND	SPLIT	SPLIT	STAND	STAND
Pair 8	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT
Pair 7	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	HIT	HIT	HIT	HIT
Pair 6	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	HIT	HIT	HIT	HIT	HIT
Pair 5	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	HIT	HIT
Pair 4	HIT	HIT	HIT	SPLIT	SPLIT	HIT	HIT	HIT	HIT	HIT
Pair 3	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	HIT	HIT	HIT	HIT
Pair 2	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	HIT	HIT	HIT	HIT

Reward Function

- Defines the goal in a RL problem
- Maps each perceived state (or state-action pair) to a single number (reward)
- Reward comes from environment, not agent
- Return: sum of agent rewards
- RL agent's goal is to maximize return
- Biology: Reward = Pleasure / Pain

Reward Function Examples

- Path-Finding
  - Goal is terminal state
  - Non-goal states have -1 reward
  - Getting to goal fastest maximizes reward
- Blackjack
  - Winning State = Positive Reward
  - Losing State = Negative Reward
  - Reward proportional to money won/lost


lsnes rr2-β23 [bsnes v085 (Compatibility core)]
File SNES Movie Speed Tools Configure Help
TIME
342
4800
Frame: 5823 Lag: 444 Subframe: 0 Slot: 1 [Nonexistent] Speed: 100% Recording
What the model sees (White area is focused by spatial transformer)
AD+X
AD+Y
AD+X
Recieved direct reward and discounted expected future reward (gamma=0.90)
Recieved direct reward and discounted expected future reward (gamma=0.90)
12
10
8
6
4
2
0
-2
4900
4920
4940
4960
4980
State

Returns

- Agent gets a reward at each time step
  - r_t, r_{t+1}, r_{t+2}, …
- What value do we want to maximize?
- We want to maximize Expected Return $E\{G_t\}$
- Episodic Tasks:
  - Return $G_t = r_t + r_{t+1} + r_{t+2} + \ldots + r_T$
  - Where T is the final time step at a terminal state
- An optimal policy π^* maximizes expected return

Value Function

- Rewards: what are immediately good
- Value functions indicate what may be eventually be good or bad (on expectation)
- Value of a state or (s,a) is the total amount of reward an agent can expect to accumulate in the future given that it is currently in this state
- Values indicate long-term desirability

Value Function

- Reward: Pleasure or Pain
- Value: How pleased or displeased we are to be here
- We seek actions that have the highest value, those will bring the highest eventual return
- Value is harder to determine than reward, so we must calculate and estimate them
- Our policy should ideally have us take high-valued actions for each possible state

Value Function Examples

- Path-Finding
  - Non-goal state on the best path to the goal has high desirability, and a high value
  - Actions taking us on the path have high value
- Blackjack
  - Me having 20 with dealer showing 6 has high probability of me winning, so a high value

Model of Environment

- Mimics the behaviour of environment
- Ex: Given state and action, the model may attempt to predict the next state
- Models are used for planning (A*, AB)
- Not necessary for RL in general, but often used in games (self play, etc)

Exploration vs. Exploitation

- One of the main challenges in RL is the trade-off between exploration and exploitation
- To obtain a lot of reward, agents must prefer actions that it knows produce good results
- In order to learn which actions produce good rewards, it must try them out first
- The agent must exploit knowledge it has, but also explore in order to gain more knowledge
- Also one of the main challenges of *real life*

Exploration vs. Exploitation Examples

- Choosing a Restaurant
  - Go to the place you know that’s alright, or try a new place you’ve never eaten at before?
- Playing Games
  - I chose door #3 last time and it had 10 gold pieces. Do I choose it next time or try another one to see if it contains more?