

CSE 417T

Introduction to Machine Learning

Lecture 23

Instructor: Chien-Ju (CJ) Ho

Logistics

- Homework 5 is due Apr 19 (Tuesday)
- Exam 2 will be on April 28 (Thursday)
 - Will focus on the topics in the second half of the semester
 - Format / logistics will be similar with what we have in Exam 1
 - Timed exam (75 min) during lecture time in the classroom
 - Closed-book exam with 2 letter-size cheat sheets allowed (4 pages in total)
 - No format limitations (it can be typed, written, or a combination)
 - April 26 (Tuesday) will be a review lecture

Recap

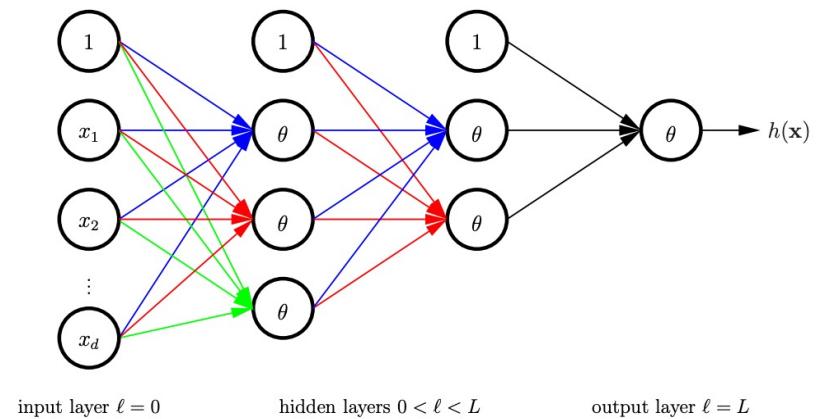
Neural Network

- Evaluate $h(\vec{x})$ given h (characterized by $\{w_{i,j}^{(\ell)}\}$)
 - Forward propagation

$$\mathbf{x} = \mathbf{x}^{(0)} \xrightarrow{\text{w}^{(1)}} \mathbf{s}^{(1)} \xrightarrow{\theta} \mathbf{x}^{(1)} \xrightarrow{\text{w}^{(2)}} \mathbf{s}^{(2)} \xrightarrow{\theta} \mathbf{x}^{(2)} \dots \xrightarrow{\text{w}^{(L)}} \mathbf{s}^{(L)} \xrightarrow{\theta} \mathbf{x}^{(L)} = h(\mathbf{x}).$$

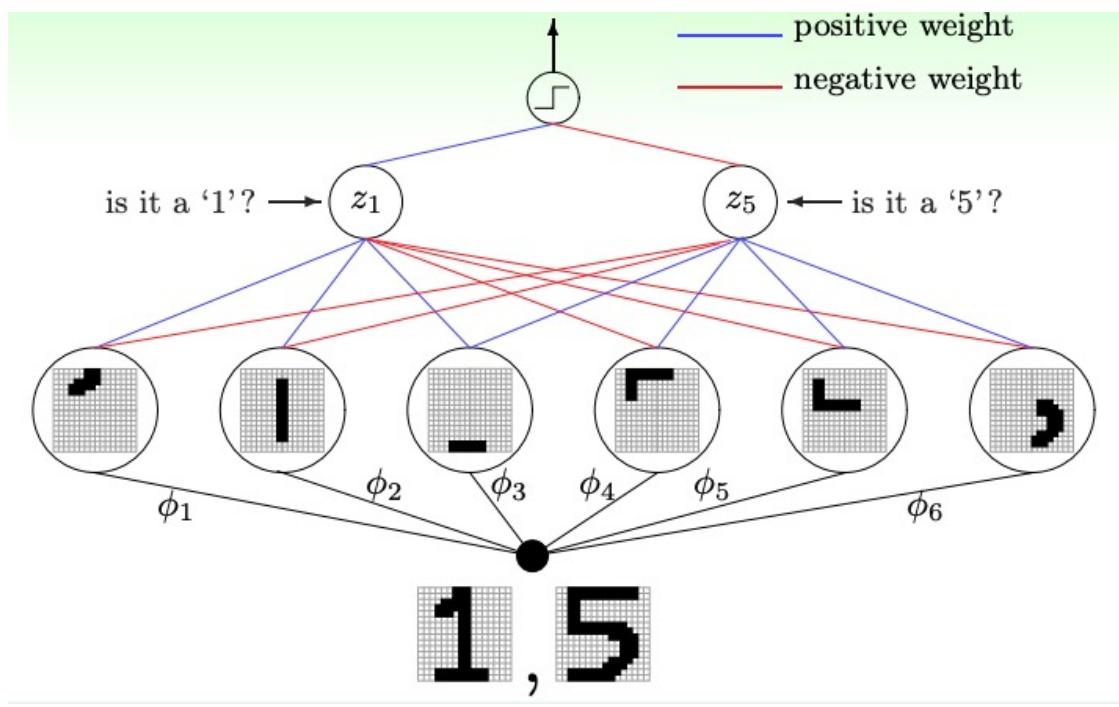
- Given D , learn the weights $W = \{w_{i,j}^{(\ell)}\}$

- Backpropagation
- Initialize $w_{i,j}^{(\ell)}$ randomly
- For $t = 1$ to T
 - Randomly pick a point from D (for stochastic gradient descent)
 - Forward propagation:** Calculate all $x_i^{(\ell)}$ and $s_i^{(\ell)}$
 - Backward propagation:** Calculate all $\delta_j^{(\ell)}$
 - Update the weights $w_{i,j}^{(\ell)} \leftarrow w_{i,j}^{(\ell)} - \eta \delta_j^{(\ell)} x_i^{(\ell-1)}$
- Return the weights



Deep Neural Network

- “Shallow” neural network is powerful (universal approximation theorem holds with a single hidden layer). Why “deep” neural networks?



Each layer captures **features** of the previous layers.

We can use “raw data” (e.g., pixels of an image) as input. The hidden layer are extracting the **features**.

Design different **network architectures** to incorporate domain knowledge.

Some Techniques in Improving Deep Learning

- Regularization to mitigate overfitting
 - Weight-based, early stopping, dropout, etc
- Incorporating domain knowledges
 - Network architectures (e.g., Convolutional Neural Nets)
- Improving computation with huge amount of data
 - Hardware architecture to improve parallel computation
- Improving gradient-based optimization
 - Choosing better **initialization** points

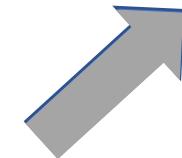
Radial Basis Functions (RBF)

- $$h(\vec{x}) = \sum_{k=1}^K w_k \phi\left(\frac{\|\vec{x} - \vec{\mu}_k\|}{r}\right)$$
- Connection to nearest neighbor
 - RBF is based on the similarity to a set of points
- Connection to linear models
 - Parametric RBF is essentially linear model with nonlinear transformation
- Connection to SVM with RBF Kernel
 - Using K representative points vs. using support vectors
- Connection to Neural Networks
 - RBF can be graphically represented as a one-hidden layer network

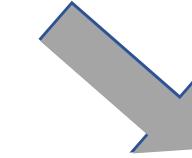
Machine Learning Lifecycle

For ML to have “positive” impacts, we need to be careful in every stage

Feedback → Task Definition



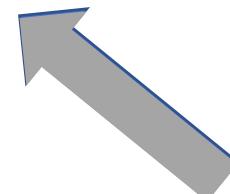
Deployment



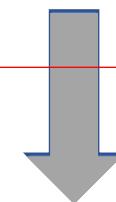
Dataset Construction

What we covered
(and majority of
ML research)

Testing



Training



Model Definition

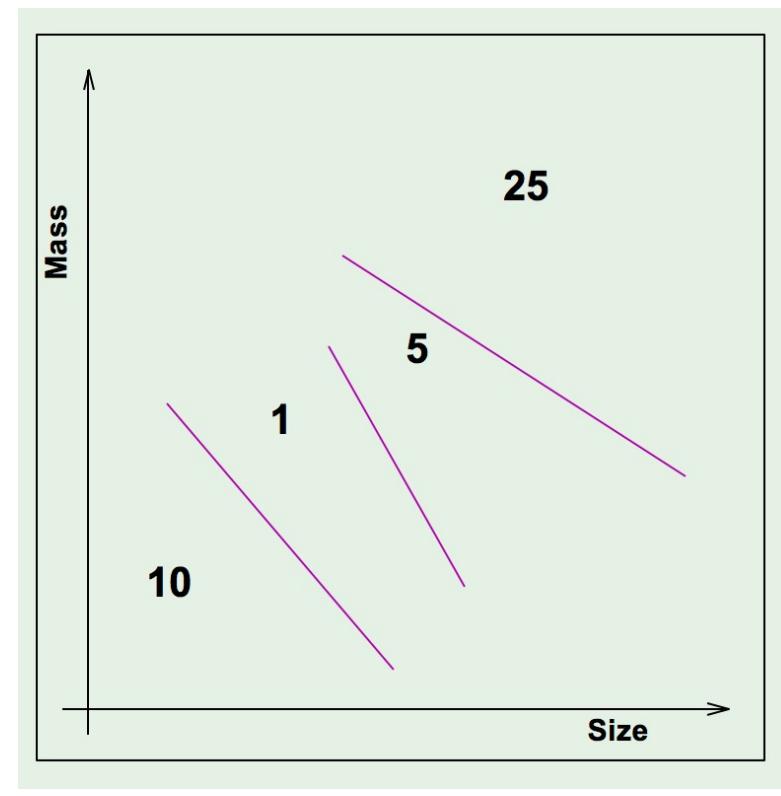
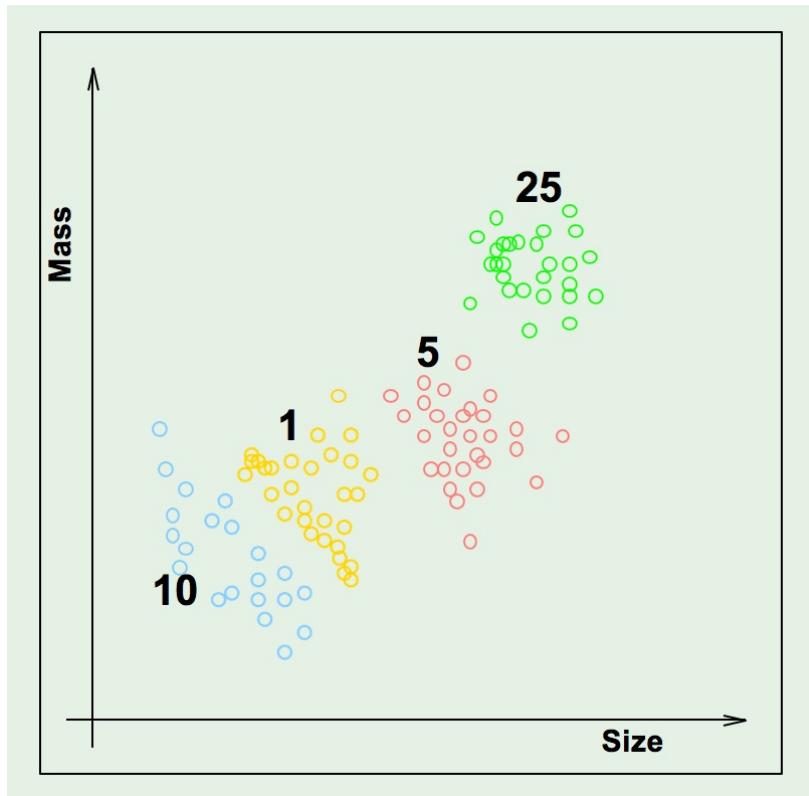
Today's Lecture

Types of Learning

- Supervised learning (the focus of this course)
 - Given training data (input, correct output)
 - Try to predict the output for data not seen before
- Unsupervised learning
 - Given data in the form of (input)
 - Find patterns in data
- Reinforcement learning
 - Learn how to act, based on rewards for actions

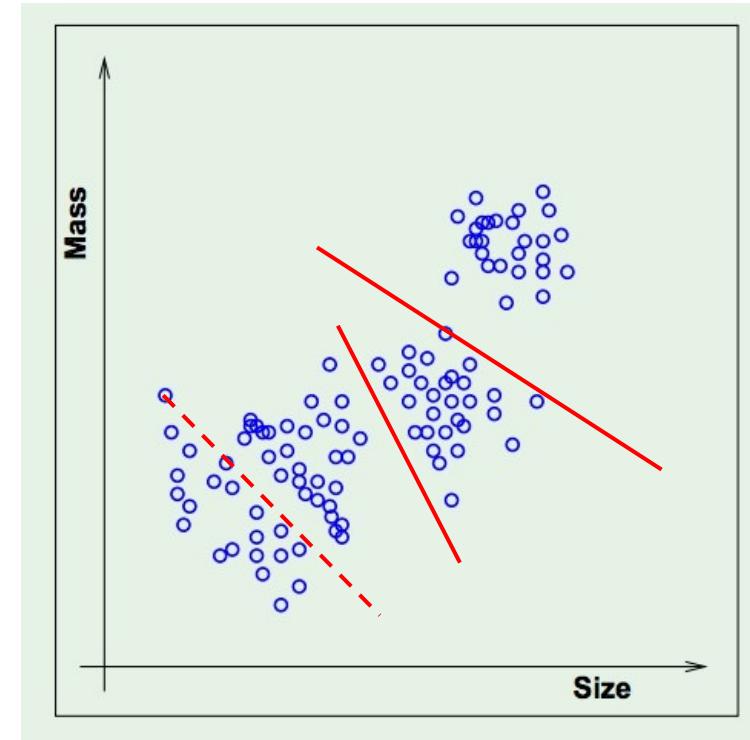
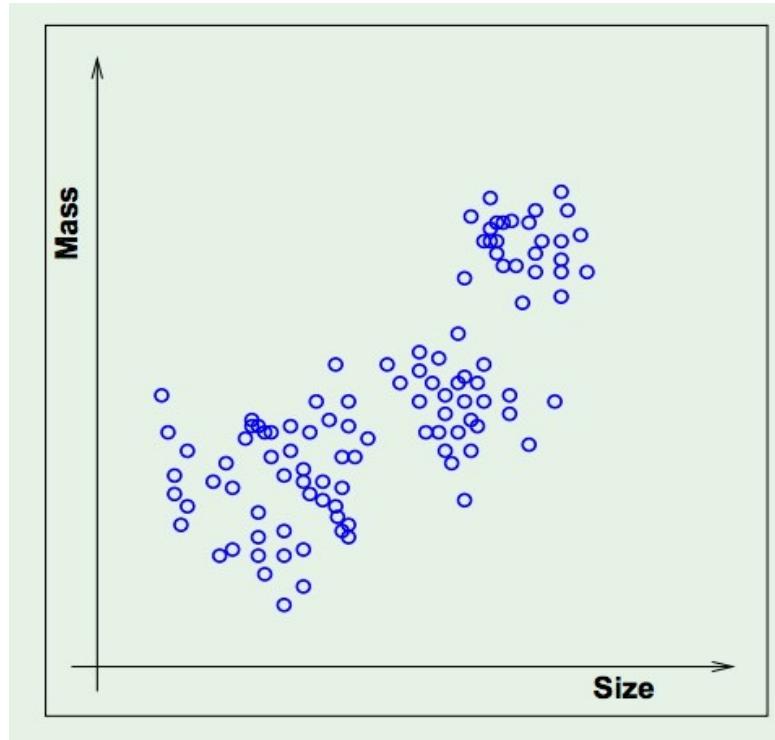
Supervised Learning

- Given data with (input, correct output), learn a pattern that can predict previously unseen data



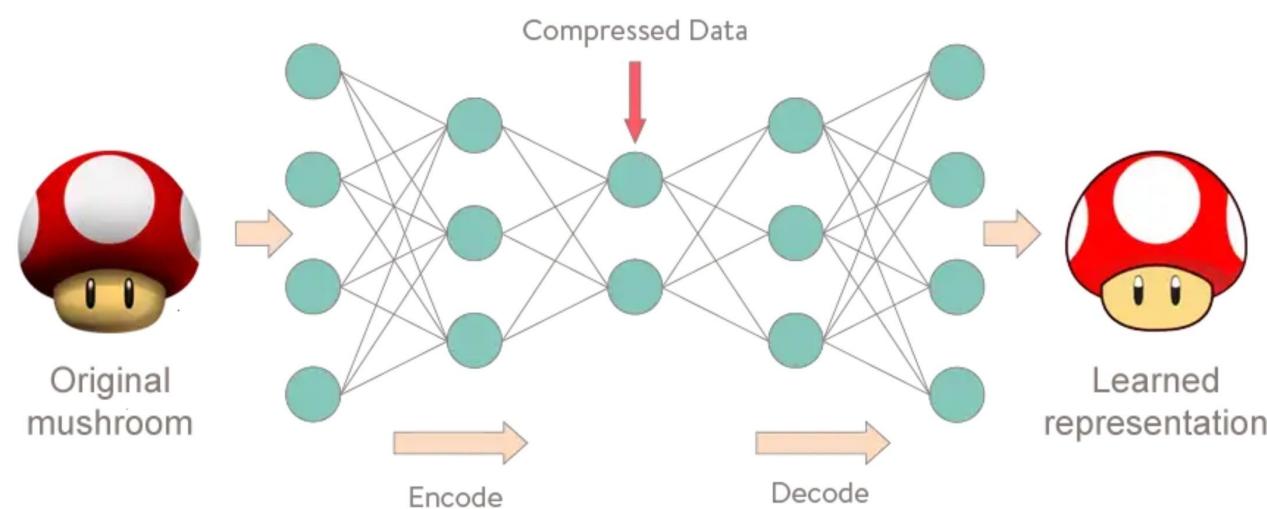
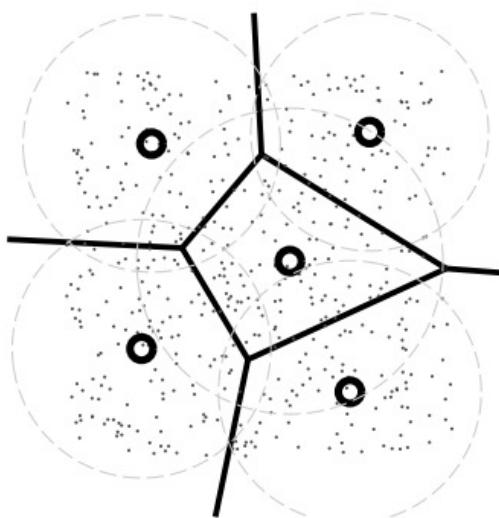
Unsupervised Learning

- Suppose you only have the feature vectors but no labels. Still want to describe the data in some useful way.



Unsupervised Learning

- Suppose you only have the feature vectors but no labels. Still want to describe the data in some useful way.



A Short Introduction of Reinforcement Learning

[The entire section of reinforcement learning is **safe to skip** for the exam]

Illustrating Applications

- Consider the following tasks
 - **News site:**
picks a news header to show to each user; don't know user preferences a priori
Goal: maximize #clicks.
 - **Dynamic pricing:**
Pick a price for your product; don't know customer information a priori
Goal: maximize total profit.
 - **Personalized health advice:**
Pick a recommendation of activity; don't know user adoption chance a priori
Goal: maximize #adopted recommendations
 - ...
- Overall goal:
 - Maximizing total rewards from actions without knowing the rewards in advance
 - Learning by doing

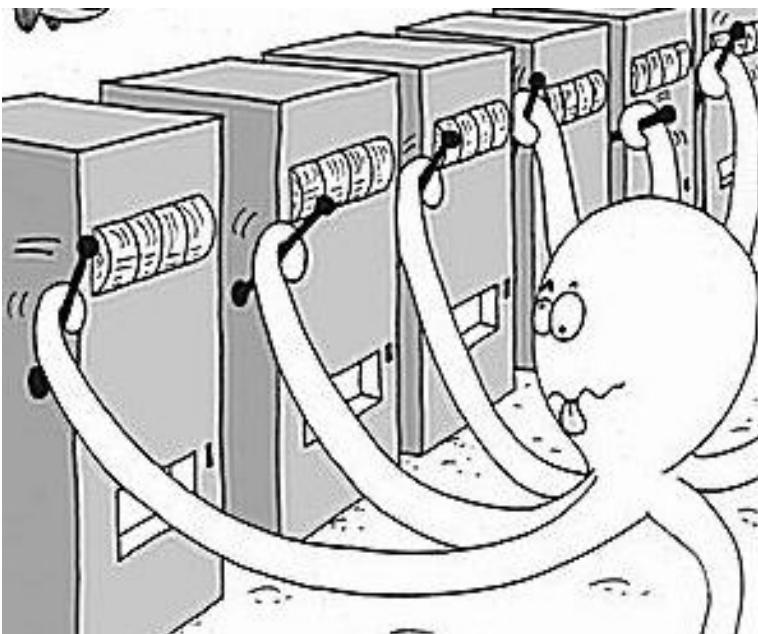
Multi-Armed Bandit

One classical formulation for reinforcement learning

One-Armed Bandit = Slot Machine



Multi-Armed Bandit



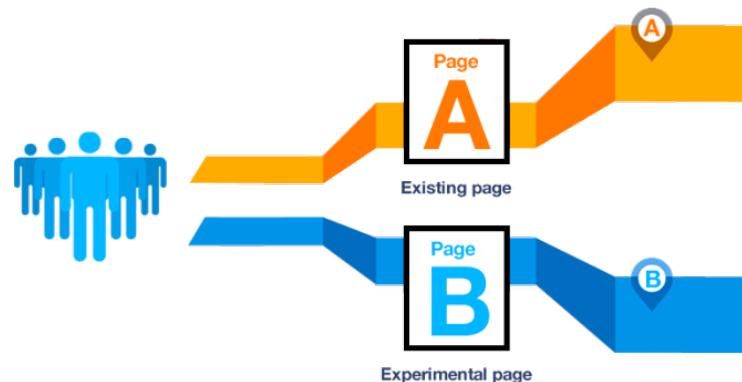
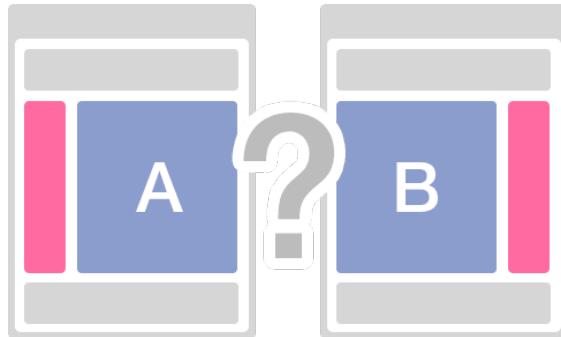
- Choose a slot machine to pull at each time
- IID Rewards: each arm reward is iid drawn from unknown distribution
- Bandit feedback: Observe only the reward of your choice
- Goal:
 - Maximize the total rewards you obtain
 - Equivalently, minimize **regret**

Regret:
 $\text{Utility(OPT)} - \text{Utility(ALG)}$

A classical framework for studying **exploration**-**exploitation** tradeoff

Example

- Which interface design should I use for my website?



- Epsilon first
 - **Exploration:** For an initial **epsilon** population, randomly split them into seeing A or B
 - **Exploitation:** Choose the one with better empirical “performance” afterwards
- How much should we explore? Can we adaptively explore?

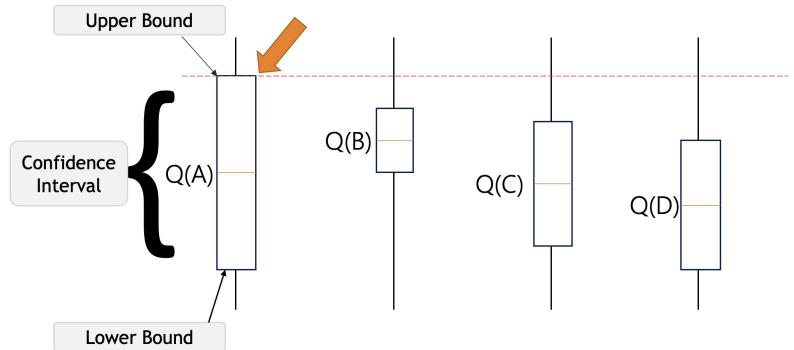
Adaptive Algorithm: Upper Confidence Bound (UCB)

- An index-based method for stochastic bandits
 - Maintain an index for each arm k at every time t
 - Select the arm with the largest index

$$I_k(t) = \bar{X}_k(t) + \sqrt{\frac{L \log t}{n_k(t)}}, \forall k.$$

Empirical mean:
exploitation

Confidence interval:
exploration

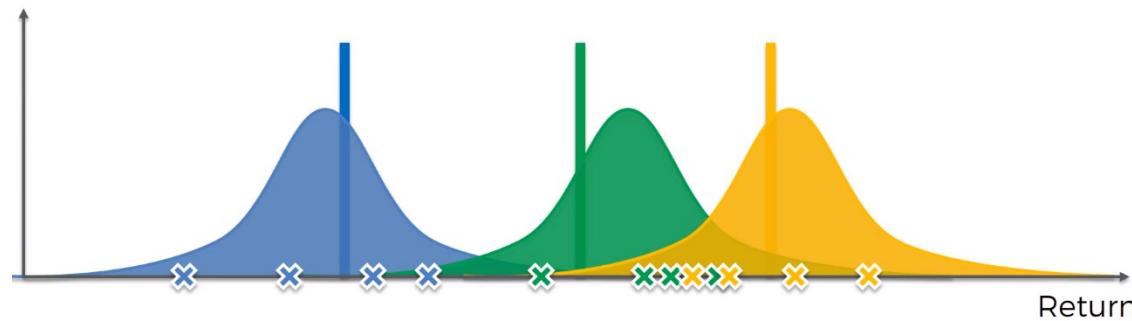


- UCB achieves **regret** bound $O(\log T)$ in stochastic settings

Common goal: Sublinear regret / no-regret:
Regret / $T \rightarrow 0$; converges to optimal action

Adaptive Algorithm: Thompson Sampling

- Thompson Sampling Algorithm
 - Maintain reward distributions of arms
 - Sample an “index” for each arm based on the distribution



- Update the distribution based on observations
- Properties:
 - Theoretically, it also achieves no regrets
 - Empirically, it usually has superior performance

Many Variations of Multi-Armed Bandits

- We might be able to observe rewards for the actions we didn't take
 - Full-information feedback vs bandit feedback
- What if rewards are not generated stochastically
 - Adversarial bandits, nonstationary bandits
- We might have additional contextual information about the rewards
 - Contextual bandits
- The rewards might depend on the “state” of the environment
 - Using Markov decision process (MDP) formulation
 - The “reinforcement learning” problem people usually refer to
 - Will talk about this next
- And more...



Made with
VivaVideo

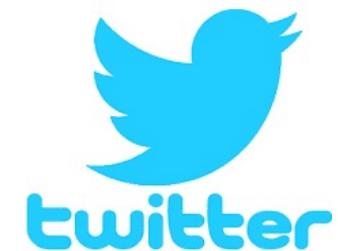
[Safe to Skip for the Exam]



- In some ways, this actually happens everyday.

Consider the slot machine as the algorithm

- Human-generated data powers the algorithm
- The algorithm influences human welfare



My Own Research: Human-in-the-Loop Bandit Learning

- **Biased** human feedback
 - The feedback received for selecting an arm might be biased by humans
- Learning as **incentives**
 - The number of possible actions could increase when humans are involved
- **Ethical** implications
 - Long-term impacts of actions in bandits
 - Privacy-preserving online learning
 - Secure sequential learning

Bandit Learning with Biased Feedback

joint work with Wei Tang

User Generated Content Platforms

YouTube search results for "arizona":
ARIZONA - Oceans Away [Official Video]
ARIZONA
1 month ago • 366,332 views
Oceans Away Available Now: Spotify:
<https://arizonar.com/oceansawayspotify>

Quora post: What is your PhD thesis in one sentence?
http://lolmythesis.com/

Richard Peng, Assistant Professor at Georgia Institute of Technology (2015-present)
Answered Jul 23, 2014 · Upvoted by Jessica Su, CS PhD student at Stanford and Karthik Abinav, PhD student in Computer Science from UMD

Viewing graphs as matrices lets you play with them as if they're positive real numbers, and leads to some really fast (parallel) algorithms for classical problems.

For reference: [Algorithm Design Using Spectral Graph Theory](#)

Abhinav Maurya
Hasn't this been a theme in computer science research for a while now? Any addition...

1,504,905 views

42K 1K

12.2k Views • 119 Upvotes

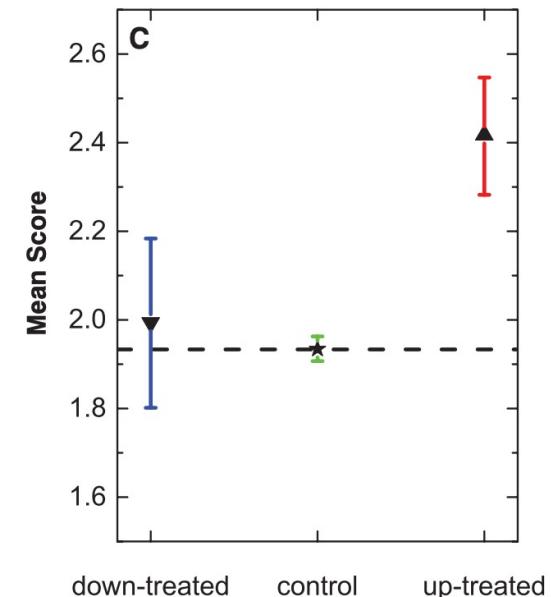
A Bandit Formulation

- When each new user arrives
 - **Exploration**: Show the user some (set of) content to obtain feedback (upvotes, likes, shares, etc)
 - **Exploitation**: Show the user the content he/she would enjoy
- Goal:
 - maximize total user happiness
 - Which may **NOT** be the same as the total number of positive feedback
- Mapping to the bandit formulation
 - Arm: the content chosen to show to users

Users' feedback might be biased



- In a Reddit-like platform, randomly insert an upvote/downvote to some posts right after they are posted.

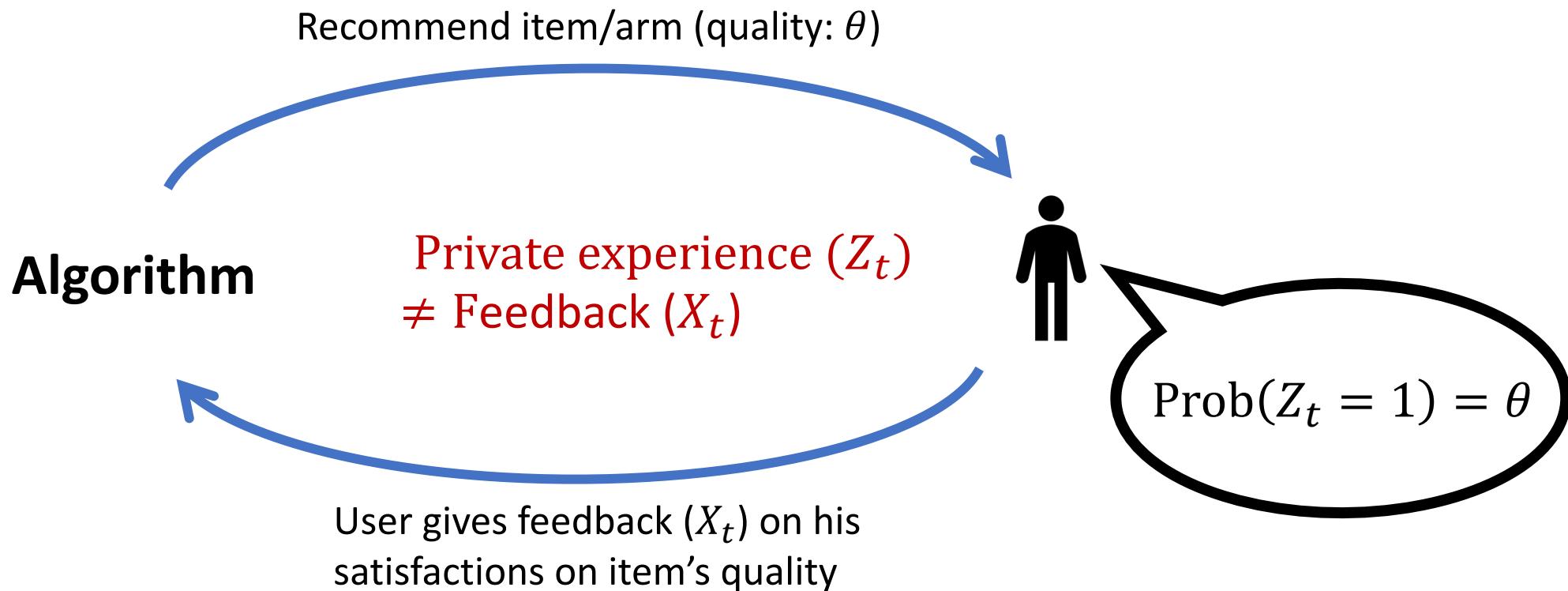


Social Influence Bias: A Randomized Experiment. Muchnik et al. Science 2013.

[Safe to Skip for the Exam]

Feedback formulation

Z_t : Whether user t likes the content
 X_t : Feedback of user t



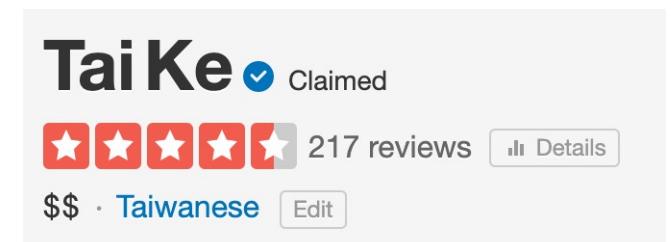
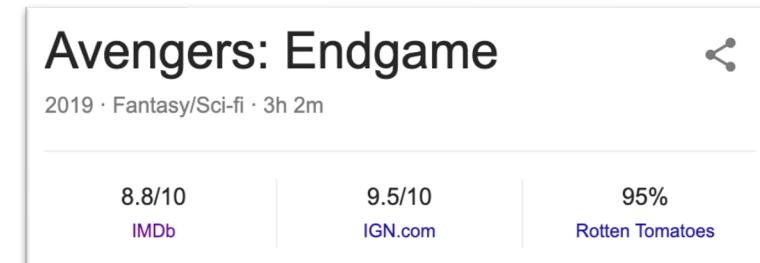
Feedback function: $\mathbb{P}(X_t = 1) = \text{Feedback}(\theta, \rho, n)$

ρ : positive votes ratio
 n : total votes received

Can we just **recover** the true latent quality from **(biased)** feedback?

Main Results

- Explore two general set of bias models
- Model 1: feedback is biased by empirical average
 - Designing no-regret bandit learning (i.e., near-optimal learning) is possible
- Model 2: feedback is biased by the whole history
 - Can't infer the latent variables
 - Impossible to achieve no-regret learning



Summary

- The models are very stylized, but we show that a small deviation of human behavior could lead to very different outcomes for machine learning.
- Implications: with human biases in data generation, sometimes even with infinitely amount of data, we won't be able to learn.

New Arm Generation in Bandit Learning: An application to User Generated Content Platforms

Joint work with Yang Liu

User-Generated Content: A Bandit Formulation

- When each new user arrives
 - **Exploration**: Show the user some (set of) content to obtain feedback (upvotes, likes, shares, etc)
 - **Exploitation**: Show the user the content he/she would enjoy
- Goal:
 - maximize total user happiness
- Mapping to the bandit formulation
 - Arm: the content chosen to show to users

Assume user feedback
is “unbiased”.

Users are Both Raters and Contributors

- When each new user arrives
 - **Exploration**: Show the user some (set of) content to obtain feedback (upvotes, likes, shares, etc)
 - **Exploitation**: Show the user the content he/she would enjoy
 - **The user decides whether to contribute new content**
- Goal:
 - maximize total user happiness
- Mapping to the bandit formulation
 - Arm: the content chosen to show to users

Assume user feedback
is “unbiased”.

Why Users Contribute?

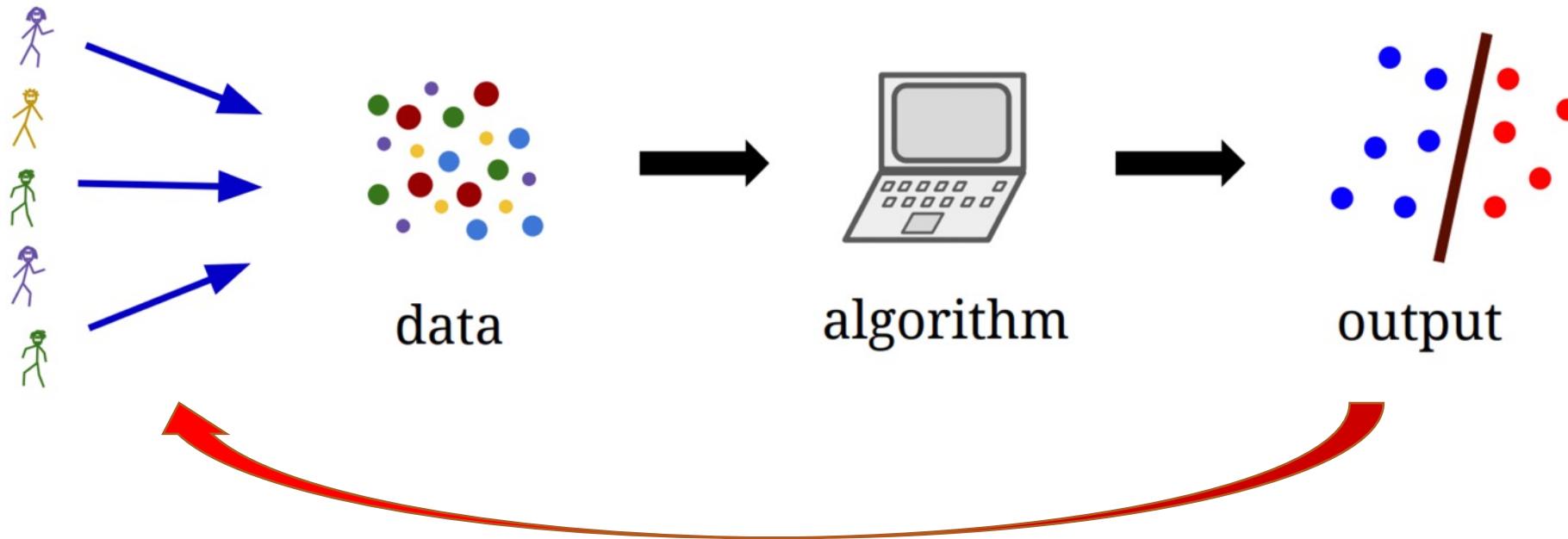
- Model:
 - Users likes attention (e.g., attention => money)



- Users aim to maximize
(Total # views of their content) – (Cost for contributing)

Need to consider **learning** and **incentives**
simultaneously when designing bandit algorithms.

A Broader Research Agenda: Strategic ML



Goodhart's law:

“If a measure becomes the public’s goal, it is no longer a good measure.”

History-Dependent Bandits and The Fairness Implications

Joint work with Wei Tang and Yang Liu

Search for “CEO” on search engines



[Kay et al., 2015]

[Safe to Skip for the Exam]

Stereotype Mirroring and Exaggeration

- Is this result mirroring the real statistics or an exaggeration?



- Assume this is mirroring of the real statistics, are there other concerns?
 - Are we reinforcing the stereotypes?
 - Are we being “unfair” to disadvantage groups that are mistreated in the past?

In the context of bandit learning

- For every action we take, we need to consider its long term impacts.
- Examples
 - **Loan applications:**
Give loans to disadvantage groups could improve their social status in the future.
 - **College admission:**
The long debate about affirmative action.
 - **Labor markets:**
Workers might get bored if kept assigned the same type of tasks.
 - **Online advertisements:**
An advertisement might be less effective if displayed too frequently (over-exposure).

Negative results

- Deploying standard bandit algorithms leads to linear regrets.
 - Simply applying standard algorithms won't work
- Even without considering fairness issues, reinforcing stereotypes is bad for the society from the utility perspective.

What Can We Do?

- Rooney rule:
 - NFL policy - when interviewing head coaches, it is required to include ethnic-minority candidates.
 - We need to give each “arm” a chance since they might be mis-treated in the past
 - A special case of **affirmative action**
- There exists setting that applying Rooney rule achieves better overall utility even when each individual decision is sub-optimal.
- Main results: Designed order-optimal algorithms

Additional Notes: Achieving Fairness is Hard

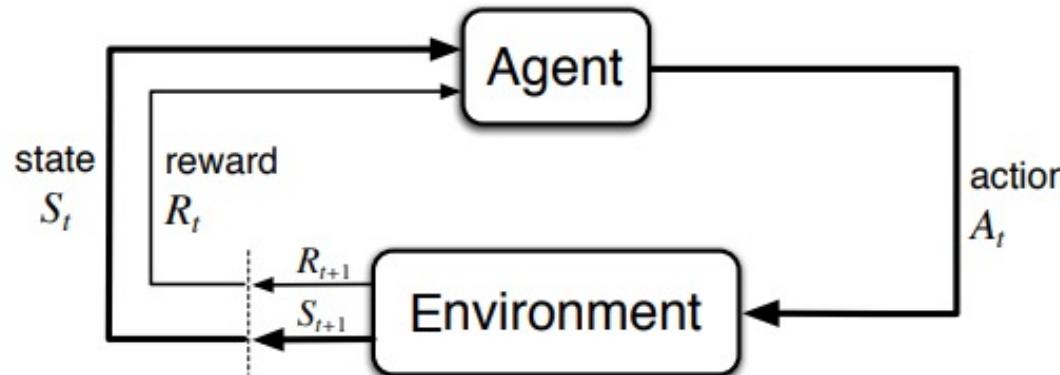
- There are many possible/reasonable fairness definitions, and they are often not compatible!
 - FAT*18 Tutorial: 21 Definitions of Fairness and Their Politics. Arvind Narayanan.
- Require conversations from people across different disciplines

Markov Decision Process (MDP)

The common formulation for reinforcement learning

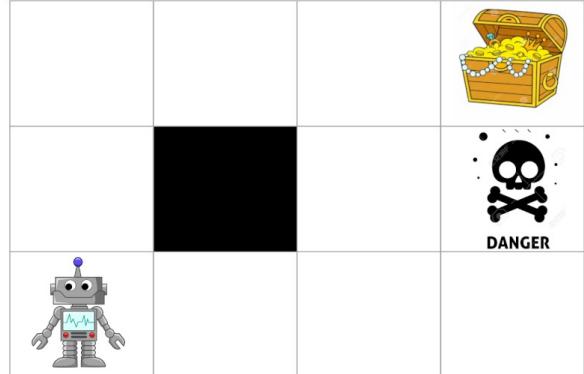
Markov Decision Process

- In multi-armed bandits, the rewards depends on **actions** only
(there are some variations to extend this idea)
- In many practical applications, the rewards depend on both the **actions** and the **state** of the environment.



Markov Decision Process

- MDP formulation
 - State space $s \in S$
 - The state of the environment, e.g., your location on the map.
 - Action space $a \in A$
 - The actions you can take, e.g., move up, move down, etc.
 - State transition $P(s', s, a)$
 - The probability of moving to state s' after taking action a in state s
 - Reward function $R(s, a)$
 - The reward obtained by taking action a in state s
- Goal:
 - Find a policy $\pi(a|s)$ (denoting the probability of taking action a in state s) that maximizes total (possibly time-discounted) rewards over time.

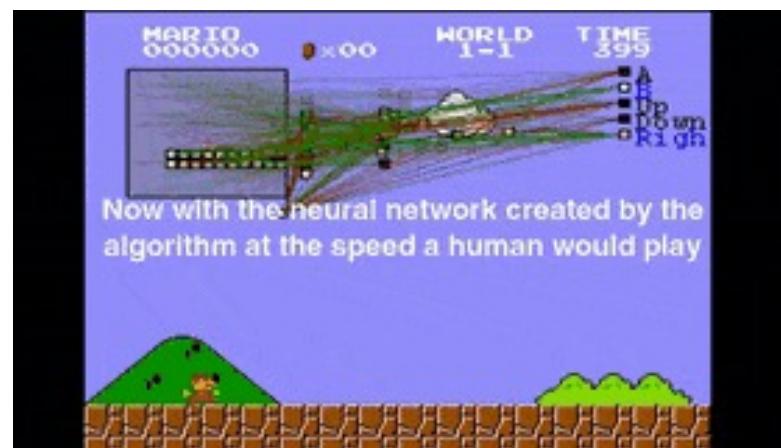
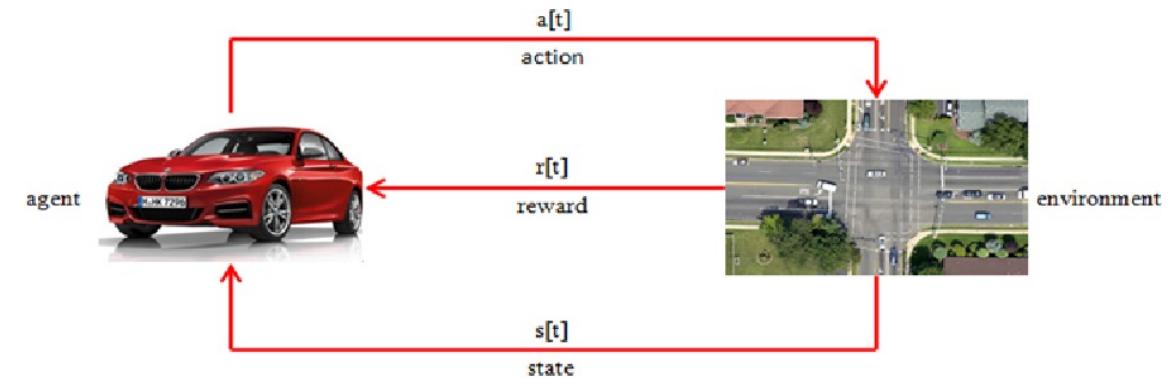
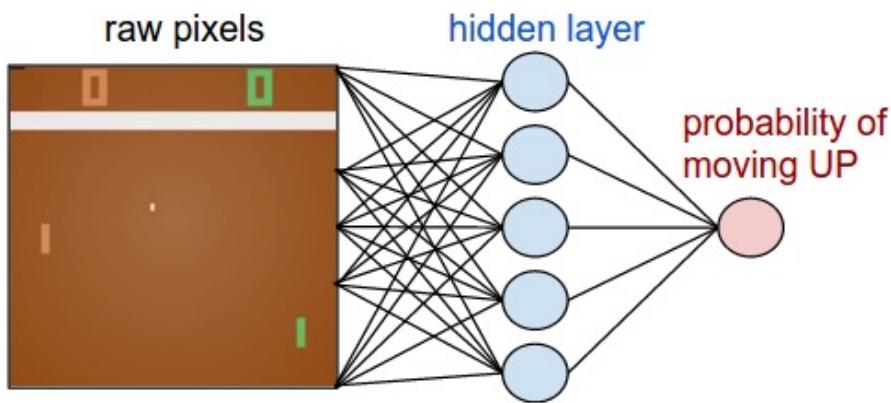


Solving MDP

- Think about the rewards of the policy $\pi(s)$
 - Value function (the value for following π starting in state s)
 - $V_\pi(s) = R(s, \pi(s)) + \gamma \sum_{s'} P(s, s', \pi(s))V_\pi(s')$
 - Q-function (the value of taking an action from a state)
 - $Q_\pi(s, a) = R(s, a) + \gamma \sum_{s'} P(s, s', a)V_\pi(s')$
 - Optimal policy
 - $\pi^*(s) = \text{argmax}_a Q_{\pi^*}(s, a)$
- Classical methods
 - Dynamic programming
 - Q learning
 - Value iteration
 - ...
- Deep reinforcement learning
 - Use neural networks to approximate important functions; apply deep learning through sampling
 - Deep Q-learning
 - Policy Gradient

[Safe to Skip for the Exam]

Rich Space of Studies



[Safe to Skip for the Exam]

Next Lecture

