# CS 760: Machine Learning
# **ML Overview**

Misha Khodak

University of Wisconsin-Madison

8 September 2025

# Announcements

- **Enrollment:**
  - Finalized this Friday. Please keep checking your status.
- **Homework:**
  - HW0 available on Canvas
  - Due Wednesday next week before class.
- **TA Office Hours:**
  - Haotian Ma: Fridays 2-3 in MH 2513
  - Avi Trost: Wednesdays 3:30-4:30 in MH 2513

# Announcements

- **Background knowledge:**
  - HW0 is for calibrating the background. Please have a look before the add/drop deadline.
  - We will assume knowledge of calculus, linear algebra, probability/statistics, etc.
  - If you consider yourself to be mathematically inclined, you should be fine.
  - Some programming experience is necessary.
    - We will use Python + some relevant libraries (e.g. Numpy).
    - Basic background on Python is needed, but no need for specific libraries.

# Outline

- **Review from last time**
  - Supervised, Unsupervised, Reinforcement Learning
- **Supervised Learning concepts**
  - Features, models, training, other terminology
- **Unsupervised Learning concepts**
  - Clustering, dimensionality reduction
- **Reinforcement Learning concepts**
  - Credit-assignment, Exploration vs. Exploitation

# Today's Learning Outcomes

- **After today's lecture:**
  - You will be able to explain the key aspects of a **supervised learning** problem.
  - Provide examples of **unsupervised learning** problems and explain why these are not supervised learning problems.
  - Explain key challenges of **reinforcement learning** problems.

# Outline

- **Review from last time**
  - Supervised, Unsupervised, Reinforcement Learning
- **Supervised Learning concepts**
  - Features, models, training, other terminology
- **Unsupervised Learning concepts**
  - Clustering, dimensionality reduction
- **Reinforcement Learning concepts**
  - Credit-assignment, Exploration vs. Exploitation

# **Review: ML Overview**: Definition

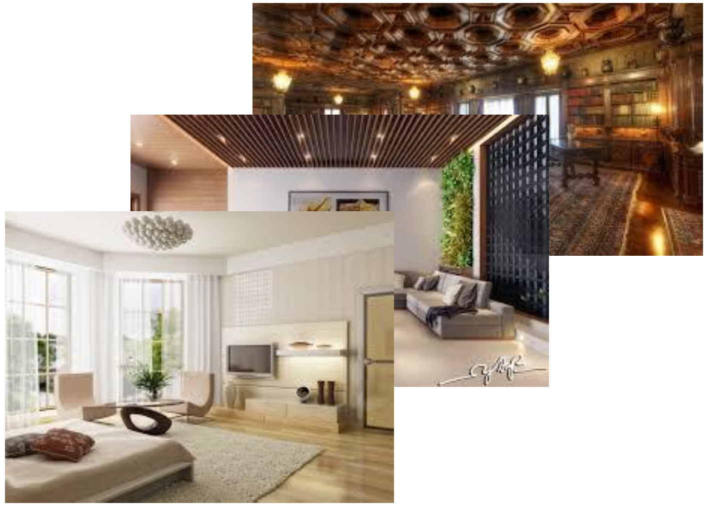
What is machine learning?

“A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P**, if its performance at tasks in **T** as measured by **P**, improves with experience **E**.” *Machine Learning*, Tom Mitchell, 1997

# **ML Overview**: Flavors

## Supervised Learning

- Learning from examples ***with* “answers”**
- **Workflow**:
  - Collect a set of examples {data, labels}: **training set**
  - “**Train**” a model to match these examples
  - “**Test**” it on new data
- **Image classification**:



indoor

outdoor

# **ML Overview**: Flavors

## Supervised Learning

- **Example: Image classification**
- Recall **T**ask/**P**erformance measure/**E**xperience definition
  - **T**ask: distinguish **indoor** vs **outdoor**
  - **P**erformance measure: probability of misclassifying
  - **E**xperience: labeled examples

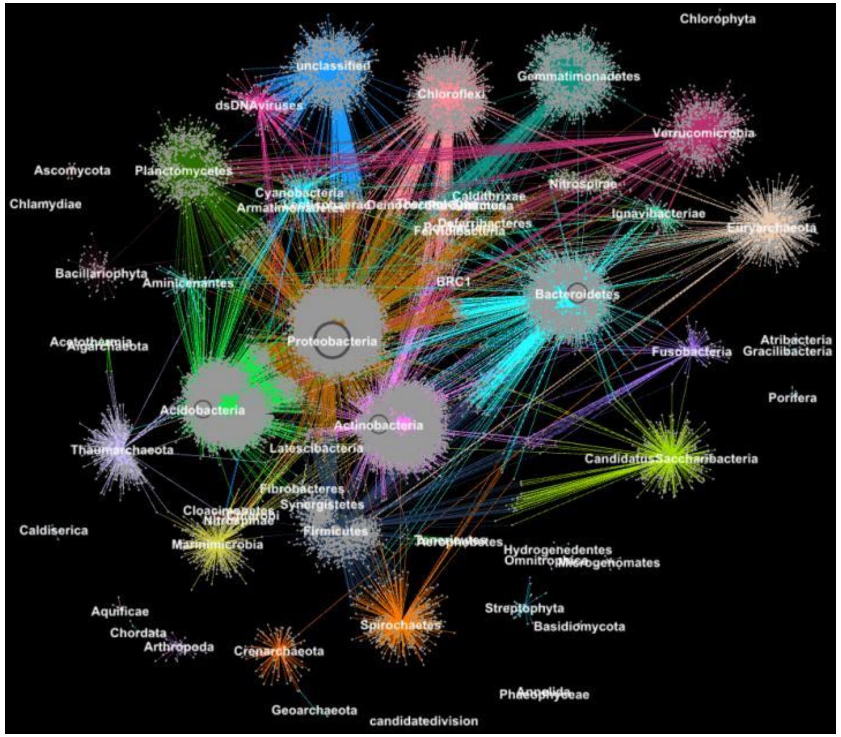
**indoor**

**outdoor**

# **ML Overview**: Flavors

## Unsupervised Learning

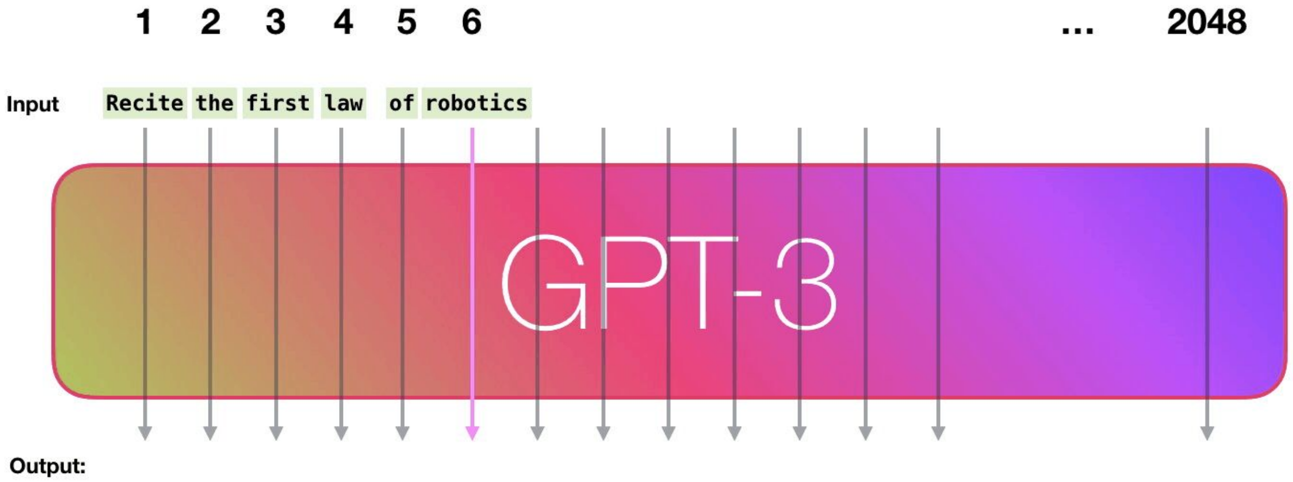
- Learning from examples ***without* "answers"**
- Goal: find some structure in the dataset
- **Workflow**:
  - Collect a set {data}
  - Perform some algorithm on it and draw insights about data
  - Sometimes: test on new data



# **ML Overview**: Flavors

## Unsupervised Learning

- **Example: Language Models**
- Recall **T**ask/**P**erformance measure/**E**xperience definition
  - **T**ask: next token (word) prediction
  - **P**erformance measure: perplexity (uncertainty or "confusion" of predicting the next word in a sequence)
  - **E**xperience: large and diverse text datasets



# **ML Overview**: Flavors

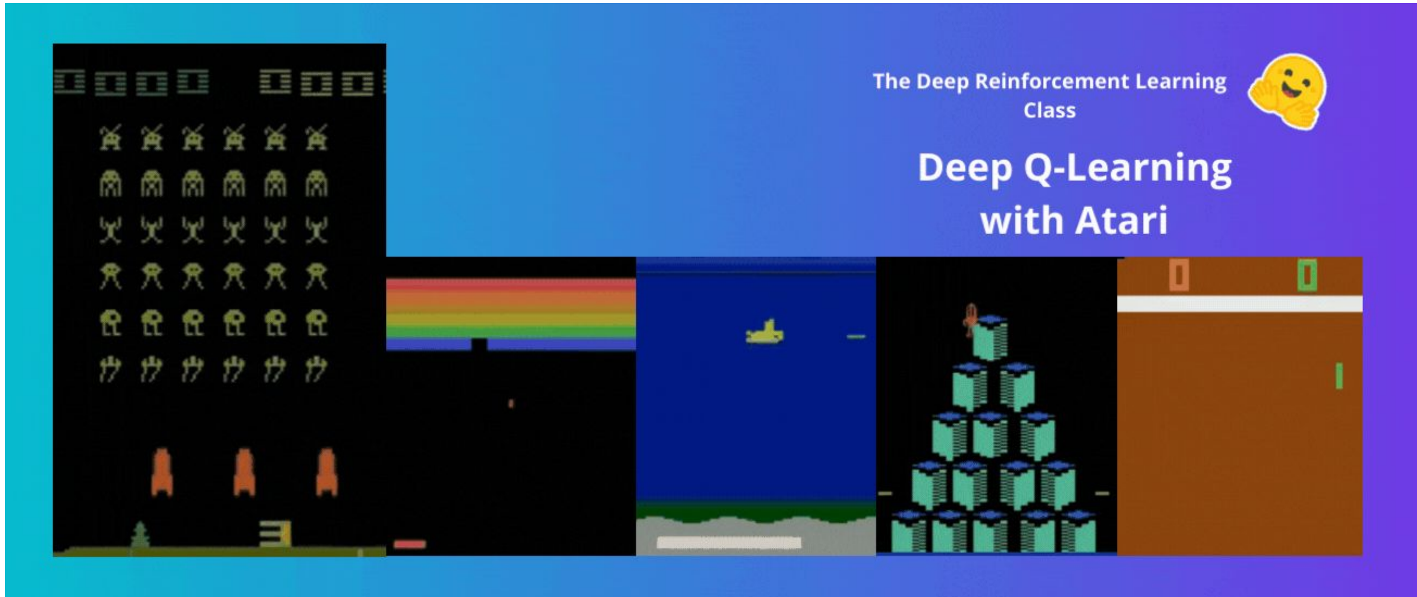
## Reinforcement Learning

- Agent interacting with the world; gets rewards for actions
- Goal: learn to perform some activity with high reward
- **Workflow**:
  - Create an environment, reward, agent
  - **Train**: train policy to maximize rewards
  - **Deploy** in new environment

# **ML Overview**: Flavors

## Reinforcement Learning

- **Example: Playing video games**
- Recall **T**ask/**P**erformance measure/**E**xperience definition
  - **T**ask: play Atari arcade games
  - **P**erformance measure: winning/advancing
  - **E**xperience: state/action/reward from previous gameplay episodes


The Deep Reinforcement Learning Class
Deep Q-Learning with Atari

# Break & Quiz

# Q1: Which of the following is generally NOT a supervised learning task?

a) Binary classification

b) Email spam detection

c) Handwriting recognition

d) Eigenvalue calculation

# Q1: Which of the following is generally NOT a supervised learning task?

a) Binary classification

b) Email spam detection

c) Handwriting recognition

**d) Eigenvalue calculation**

**Eigenvalue calculation is a mathematical problem, and we do not have any labels for this problem.**

# Outline

- **Review from last time**
  - Supervised, Unsupervised, Reinforcement Learning
- **Supervised Learning concepts**
  - Features, models, training, other terminology
- **Unsupervised Learning concepts**
  - Clustering, anomaly detection, dimensionality reduction
- **Reinforcement Learning learning concepts**
  - Exploration vs. Exploitation, credit-assignment.

# Supervised Learning

- Can I eat this?

- Safe or poisonous?
  - **Never seen it before**

- How to decide?

# **Supervised Learning:** Training Instances

- I know about other mushrooms:

safe

poisonous

- Training set of **examples/instances/labeled data**

# **Supervised Learning**: Formal Setup

## Problem setting

- Set of possible instances $\mathcal{X}$ Set of outputs $\mathcal{Y}$
- Unknown *target function* $f : \mathcal{X} \to \mathcal{Y}$
- Set of *models* (a.k.a. *hypotheses*): $\mathcal{H} = \{h | h : \mathcal{X} \to \mathcal{Y}\}$
- Training set of instances for unknown target function,

$$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})$$

safe

poisonous

safe

# **Supervised Learning**: Formal Setup

**Problem setting**

- Set of possible instances $\mathcal{X}$ Set of outputs $\mathcal{Y}$
- Unknown *target function* $f : \mathcal{X} \to \mathcal{Y}$
- Set of *models* (a.k.a. *hypotheses*): $\mathcal{H} = \{h | h : \mathcal{X} \to \mathcal{Y}\}$
- Training set of instances for unknown target function,

$$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})$$

**Goal**: model *h* that best approximates *f*

# **Supervised Learning**: Objects

**Three types of sets**

- Input space, output space, hypothesis class

$$\mathcal{X}, \mathcal{Y}, \mathcal{H}$$

- **Examples**:
  - Input space: feature vectors $\mathcal{X} \subseteq \mathbb{R}^d$
  - Output space:
    - **Binary classification** $\mathcal{Y} = \{-1, +1\}$
    - **Continuous** $\mathcal{Y} \subseteq \mathbb{R}$

safe poisonous

# **Input Space:** Feature Vectors

- Need a way to represent instance information (no need to use raw image):

$$\begin{array}{cccccccc} & & \text{cap-shape} & \text{cap-surface} & \text{cap-color} & \text{bruises} & \text{odor} & \\ \mathbf{x}^{(1)} = & \langle & \text{bell}, & \text{fibrous}, & \text{gray}, & \text{false}, & \text{foul}, \square & \rangle \end{array}$$

safe

- For each instance, store features as a vector.
  - **Next: What kinds of features can we have?**

# **Input Space**: Feature Types

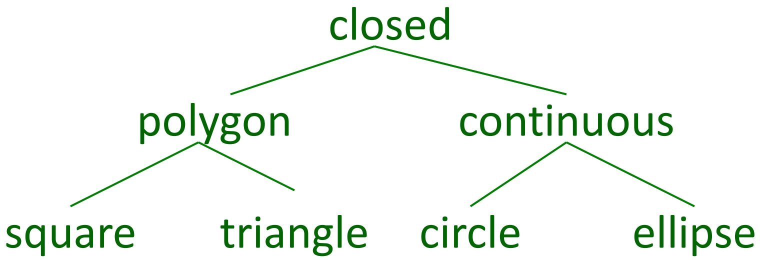
- *nominal* (including Boolean)
  - no ordering among values (e.g. *color* $\in$ {*red*, *blue*, *green*} (vs. *color = 1000* Hertz))

- *ordinal*
  - values of the feature are totally ordered (e.g. *size* $\in$ {*small, medium, large*})

- *numeric (*continuous*)*

  *height* $\in$ [0, 100] inches

- *hierarchical*
  - possible values are partially *ordered* in a hierarchy, e.g. *shape*



closed

polygon

continuous

square

triangle

circle

ellipse

# **Input Space**: Features Example

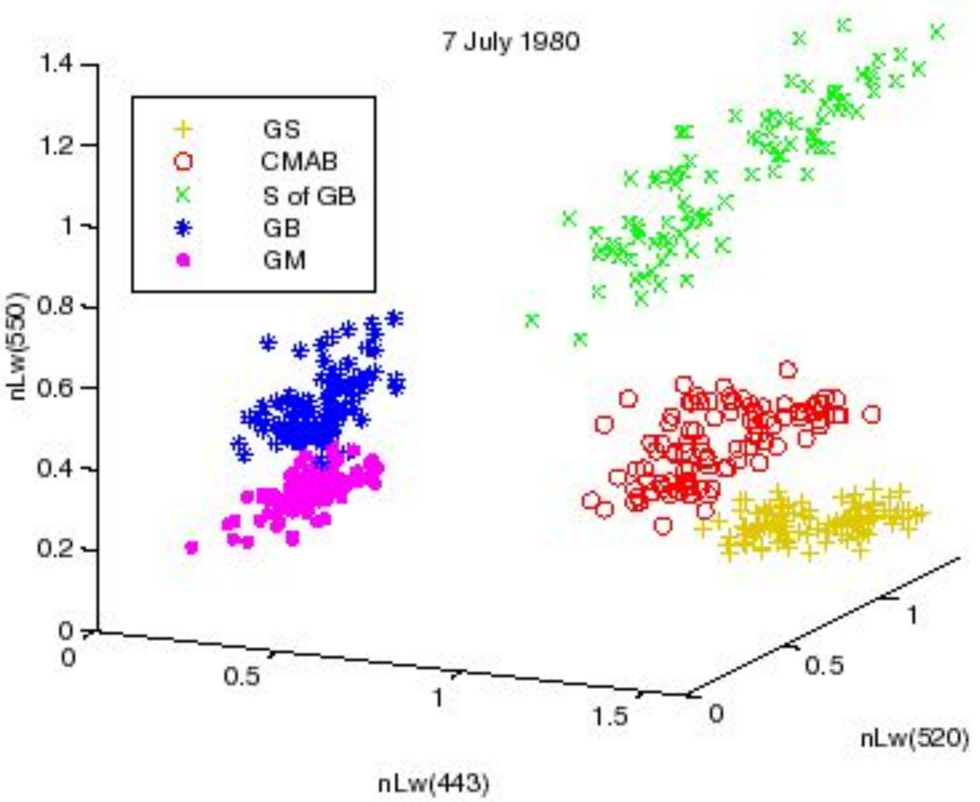
***sunken* is one possible value of the *cap-shape* feature**

**Mushroom features (UCI Repository)**

cap-shape: bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s
cap-surface: fibrous=f,grooves=g,scaly=y,smooth=s
cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r, pink=p,purple=u,red=e,white=w,yellow=y
bruises?: bruises=t,no=f
odor: almond=a,anise=l,creosote=c,fishy=y,foul=f, musty=m,none=n,pungent=p,spicy=s
gill-attachment: attached=a,descending=d,free=f,notched=n
gill-spacing: close=c,crowded=w,distant=d
gill-size: broad=b,narrow=n
gill-color: black=k,brown=n,buff=b,chocolate=h,gray=g, green=r,orange=o,pink=p,purple=u,red=e, white=w,yellow=y
stalk-shape: enlarging=e,tapering=t
stalk-root: bulbous=b,club=c,cup=u,equal=e, rhizomorphs=z,rooted=r,missing=?
stalk-surface-above-ring: fibrous=f,scaly=y,silky=k,smooth=s
stalk-surface-below-ring: fibrous=f,scaly=y,silky=k,smooth=s
stalk-color-above-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y
stalk-color-below-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y
veil-type: partial=p,universal=u
veil-color: brown=n,orange=o,white=w,yellow=y
ring-number: none=n,one=o,two=t
ring-type: cobwebby=c,evanescent=e,flaring=f,large=l, none=n,pendant=p,sheathing=s,zone=z
spore-print-color: black=k,brown=n,buff=b,chocolate=h,green=r, orange=o,purple=u,white=w,yellow=y
population: abundant=a,clustered=c,numerous=n, scattered=s,several=v,solitary=y
habitat: grasses=g,leaves=l,meadows=m,paths=p, urban=u,waste=w,woods=d

# **Input Space**: Feature Spaces

- *If all features are numeric*, we can think of each instance as a point in a $d$-dimensional Euclidean feature space where $d$ is the number of features



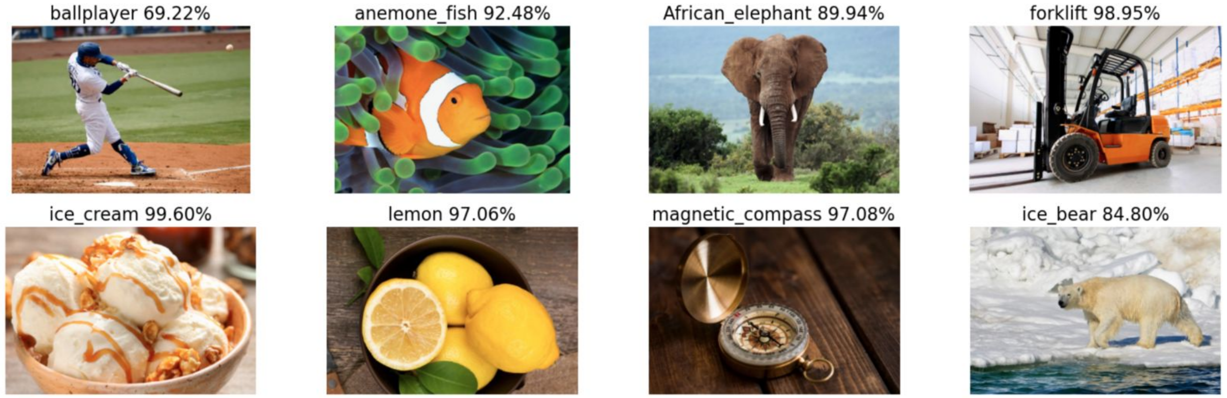
- **Example**: optical properties of oceans in three spectral bands

[Traykovski and Sosik, *Ocean Optics XIV Conference Proceedings*, 1998]

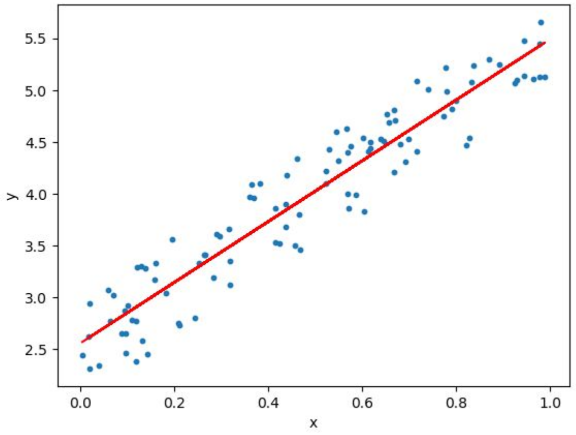
# **Output space:** Classification vs. Regression

Choices of $\mathcal{Y}$ have special names:

- Discrete: “**classification**”. The elements of $\mathcal{Y}$ are **classes**
  - Note: does not have to be binary



- Continuous: “**regression**”
  - Example: linear regression



- There are other types…

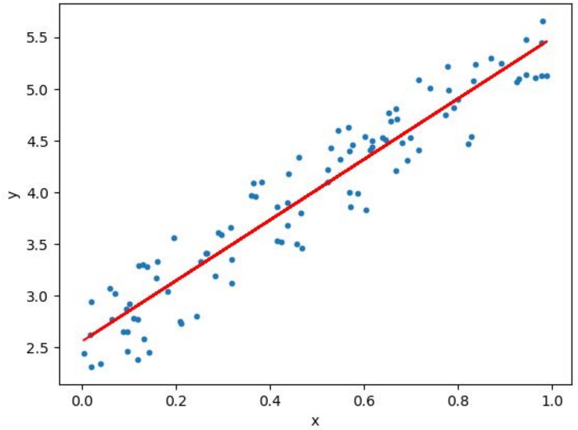
# Hypothesis class

We talked about $\mathcal{X}, \mathcal{Y}$ what about $\mathcal{H}$ ?

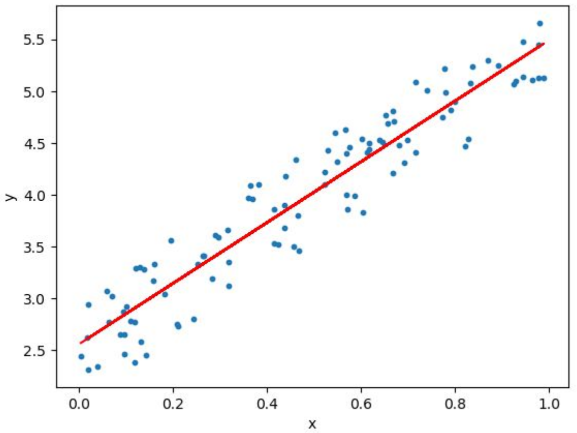
- Recall: hypothesis class / model space.
  - Theoretically, could be all maps from $\mathcal{X}$ to $\mathcal{Y}$
  - But - does not work! We'll see why later.

- Pick specific class of models. E.g. linear models:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_d x_d$$

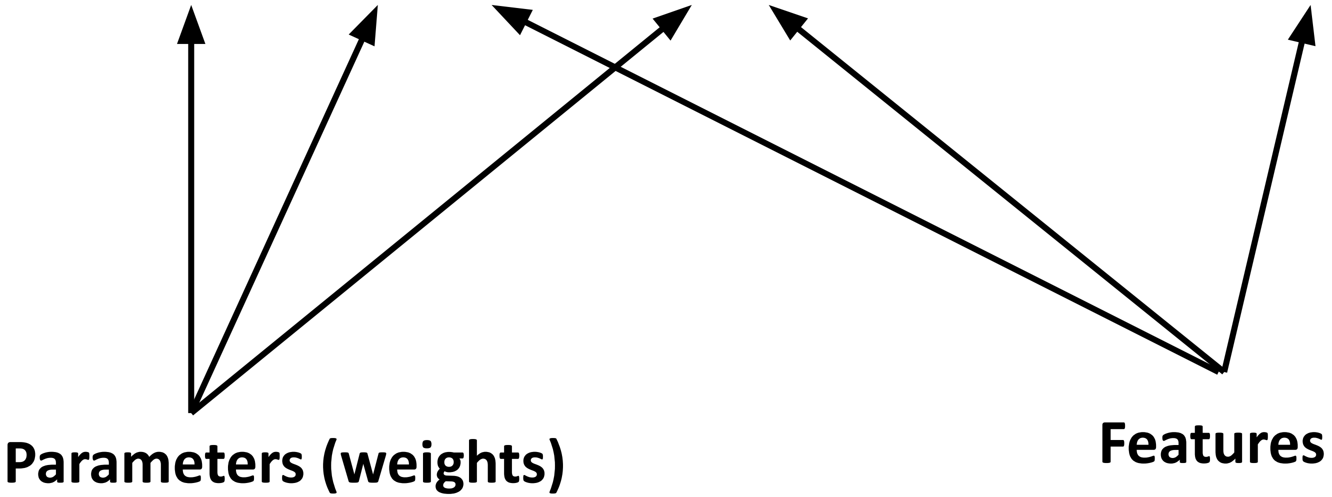


# **Hypothesis class:** Linear Functions



- **Example** class of models: linear models

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_d x_d$$



**Parameters (weights)**

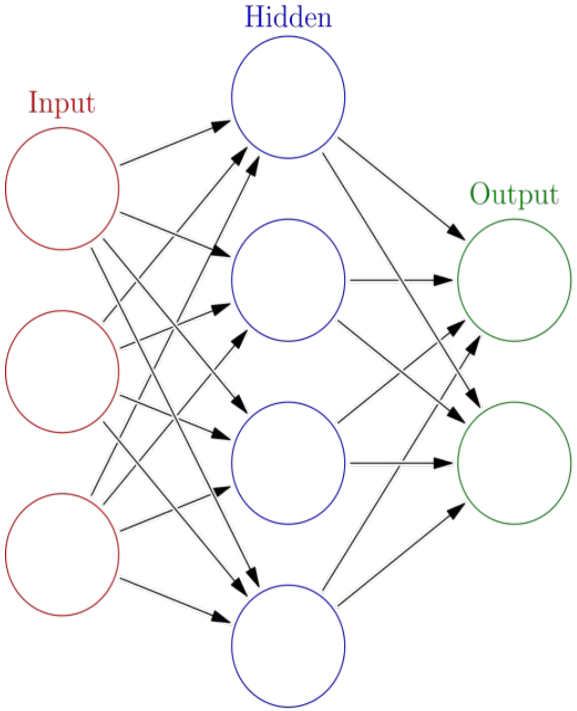
**Features**

- How many linear functions are there?
  - Can any function be fit by a linear model?

# **Hypothesis class:** Other Examples

**Example** classes of models: (deep) neural networks

$$f^{(k)}(x) = \sigma(W_k^T f^{(k-1)}(x)))$$



Wikipedia

Feedforward network

- Each layer:
  - linear transformation
  - Non-linearity

  - What are the parameters here?

# **Supervised Learning**: Formal Setup

**Problem setting**

- Set of possible instances $\mathcal{X}$ Set of outputs $\mathcal{Y}$ ✓
- Unknown *target function* $f : \mathcal{X} \rightarrow \mathcal{Y}$ ✓
- Set of *models* (a.k.a. *hypotheses*): $\mathcal{H} = \{h | h : \mathcal{X} \rightarrow \mathcal{Y}\}$ ✓
- Training set of instances for unknown target function,

$$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})$$ ✓

**Goal**: model *h* that best approximates *f*

# **Supervised Learning:** Training

**Goal:** model *h* that best approximates *f*

- One way: empirical risk minimization (ERM)

$$\hat{f} = \arg\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(h(x^{(i)}), y^{(i)}))$$

Model prediction

Hypothesis Class

Loss function: how far is the prediction from the label)?

# Batch vs. Online Learning

- **Batch learning**: get all your instances at once

$$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})$$

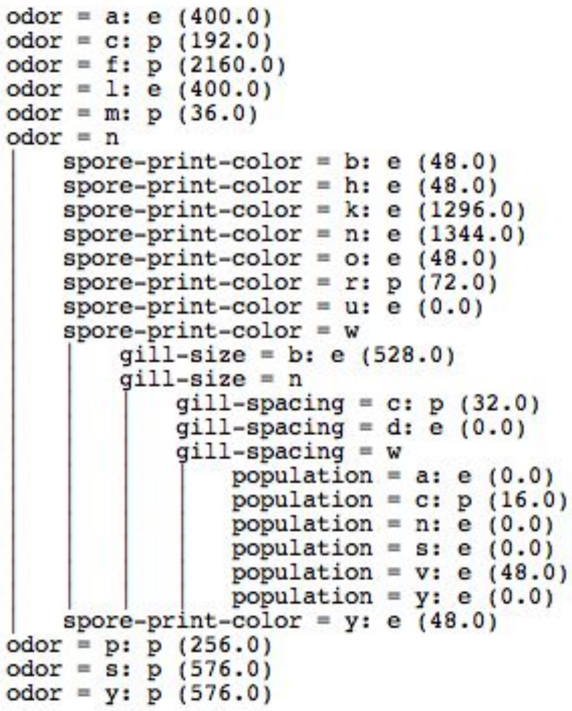
- **Online learning**: get them sequentially
  - Train a model on initial group, then update

$$\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\} \qquad \{(x^{(m+1)}, y^{(m+1)})\}$$

# **Supervised Learning:** Predicting

Now that we have our learned model, we can use it for predictions.

$\mathbf{x} = \langle \text{bell, fibrous, brown, false, foul, ...} \rangle$



```
odor = a: e (400.0)
odor = c: p (192.0)
odor = f: p (2160.0)
odor = l: e (400.0)
odor = m: p (36.0)
odor = n
|   spore-print-color = b: e (48.0)
|   spore-print-color = h: e (48.0)
|   spore-print-color = k: e (1296.0)
|   spore-print-color = n: e (1344.0)
|   spore-print-color = o: e (48.0)
|   spore-print-color = r: p (72.0)
|   spore-print-color = u: e (0.0)
|   spore-print-color = w
|   |   gill-size = b: e (528.0)
|   |   gill-size = n
|   |   |   gill-spacing = c: p (32.0)
|   |   |   gill-spacing = d: e (0.0)
|   |   |   gill-spacing = w
|   |   |   |   population = a: e (0.0)
|   |   |   |   population = c: p (16.0)
|   |   |   |   population = n: e (0.0)
|   |   |   |   population = s: e (0.0)
|   |   |   |   population = v: e (48.0)
|   |   |   |   population = y: e (0.0)
|   spore-print-color = y: e (48.0)
odor = p: p (256.0)
odor = s: p (576.0)
odor = y: p (576.0)
```

**safe or poisonous**

# Recall supervised learning workflow

- Collect a set of examples {data, labels}: **training set** ✓

- "**Train**" a model to match these examples
  - E.g. Choose a ***hypothesis class*** and perform **ERM** ✓

$$\hat{f} = \arg\min_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \ell(h(x^{(i)}), y^{(i)})$$

Model prediction

Hypothesis Class

Loss function

- **"Test"** it on new data

# Generalization

Fitting data isn't the only task, we want to **generalize**

- Apply learned model to unseen data:
  - For $(x, y) \sim \mathcal{D}$,

$$\mathbb{E}_{\mathcal{D}}[\ell(\hat{f}(x), y)]$$

- Can study theoretically or empirically
  - For theory: need assumptions, ie, training instances are iid
  - Not always the case!
    - Sequential data

# Break & Quiz

# Q2-1: Which of the following is a NOMINAL feature as introduced in the lecture?

a) Cost ∈ [0, 100]

b) Awarded ∈ {True, False}

c) Steak ∈ {Rare, Medium Rare, Medium, Medium Well, Well Done}

d) Attitude ∈ {strongly disagree, disagree, neutral, agree, strongly agree}

Q2-1: Which of the following is a NOMINAL feature as introduced in the lecture?

a) Cost ∈ [0, 100]

**b) Awarded** ∈ **{True, False}**

c) Steak ∈ {Rare, Medium Rare, Medium, Medium Well, Well Done}

d) Attitude ∈ {strongly disagree, disagree, neutral, agree, strongly agree}

# Q2-2: What is the dimension of the following feature space?

The CIFAR-10 dataset contains 60,000 32x32 **color** images in 10 different classes. (convert each data to a vector)

a) 10
b) 60,000
c) 3072
d) 1024

# Q2-2: What is the dimension of the following feature space?

The CIFAR-10 dataset contains 60,000 32x32 **color** images in 10 different classes. (convert each data to a vector)

a) 10

b) 60,000

**c) 3072** ←

d) 1024

**Every color image has 3 channels (RGB) and 32*32 pixels, so the dimension is 3*32*32=3072.**

Q2-3: Are these statements true or false?

(A) Instances from time series are independent and identically distributed (iid).

(B) The primary objective of supervised learning is to find a model that achieves the highest accuracy on the training data.

a) True, True

b) True, False

c) False, True

d) False, False

Q2-3: Are these statements true or false?

(A) Instances from time series are independent and identically distributed (iid).

(B) The primary objective of supervised learning is to find a model that achieves the highest accuracy on the training data.

a) True, True

b) True, False

c) False, True

**d) False, False**

**(A) Instances from time series usually have dependencies on the previous instances.**

**(B) The primary objective of supervised learning is to find a model that generalizes.**

# Outline

- **Review from last time**
  - Supervised, Unsupervised, Reinforcement Learning
- **Supervised Learning concepts**
  - Features, models, training, other terminology
- **Unsupervised Learning concepts**
  - Clustering, dimensionality reduction
- **Reinforcement Learning concepts**
  - Credit-assignment, Exploration vs. Exploitation

# **Unsupervised Learning:** Setup

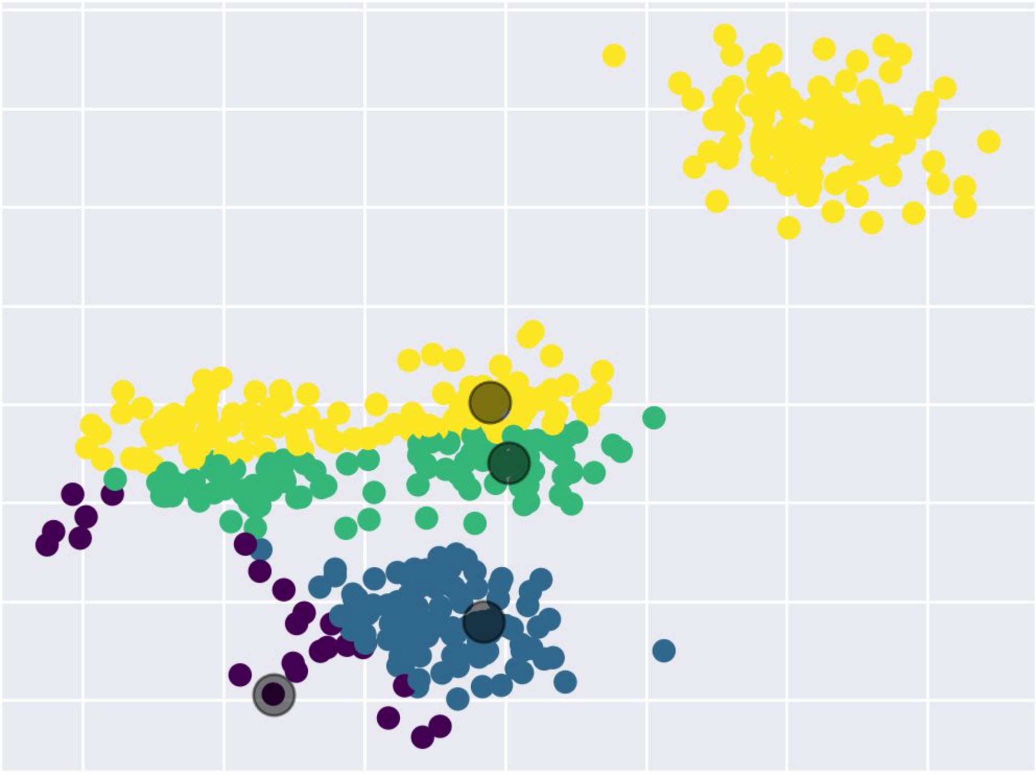
- Given instances $\{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$

- **Goal**: discover interesting regularities/structures/patterns that characterize the instances. For example:
    - **clustering**
    - **dimensionality reduction**
    - generative models, etc …

# **Clustering:** Setup

- Given instances $\{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$

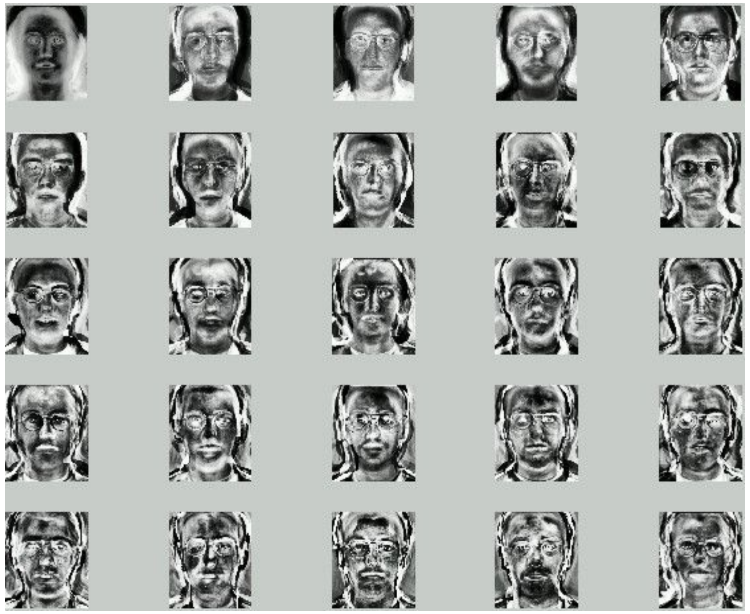
- **Goal**: model *h* divides the training set into clusters with
  - intra-cluster similarity
  - inter-cluster dissimilarity



# **Dimensionality Reduction:** Setup

- Given instances $\{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$

- **Goal**: model *h* that represents *x* with
  - lower-dim. feature vectors
  - preserving information
- Example: Eigenfaces



# **Dimensionality Reduction:** Setup

Example: Eigenfaces

$= \alpha_1^{(1)} \times$ $+ \alpha_2^{(1)} \times$ $+ \ldots + \alpha_{20}^{(1)} \times$

$$x^{(1)} = \langle \alpha_1^{(1)}, \alpha_2^{(1)}, \ldots, \alpha_{20}^{(1)} \rangle$$

$= \alpha_1^{(2)} \times$ $+ \alpha_2^{(2)} \times$ $+ \ldots + \alpha_{20}^{(2)} \times$

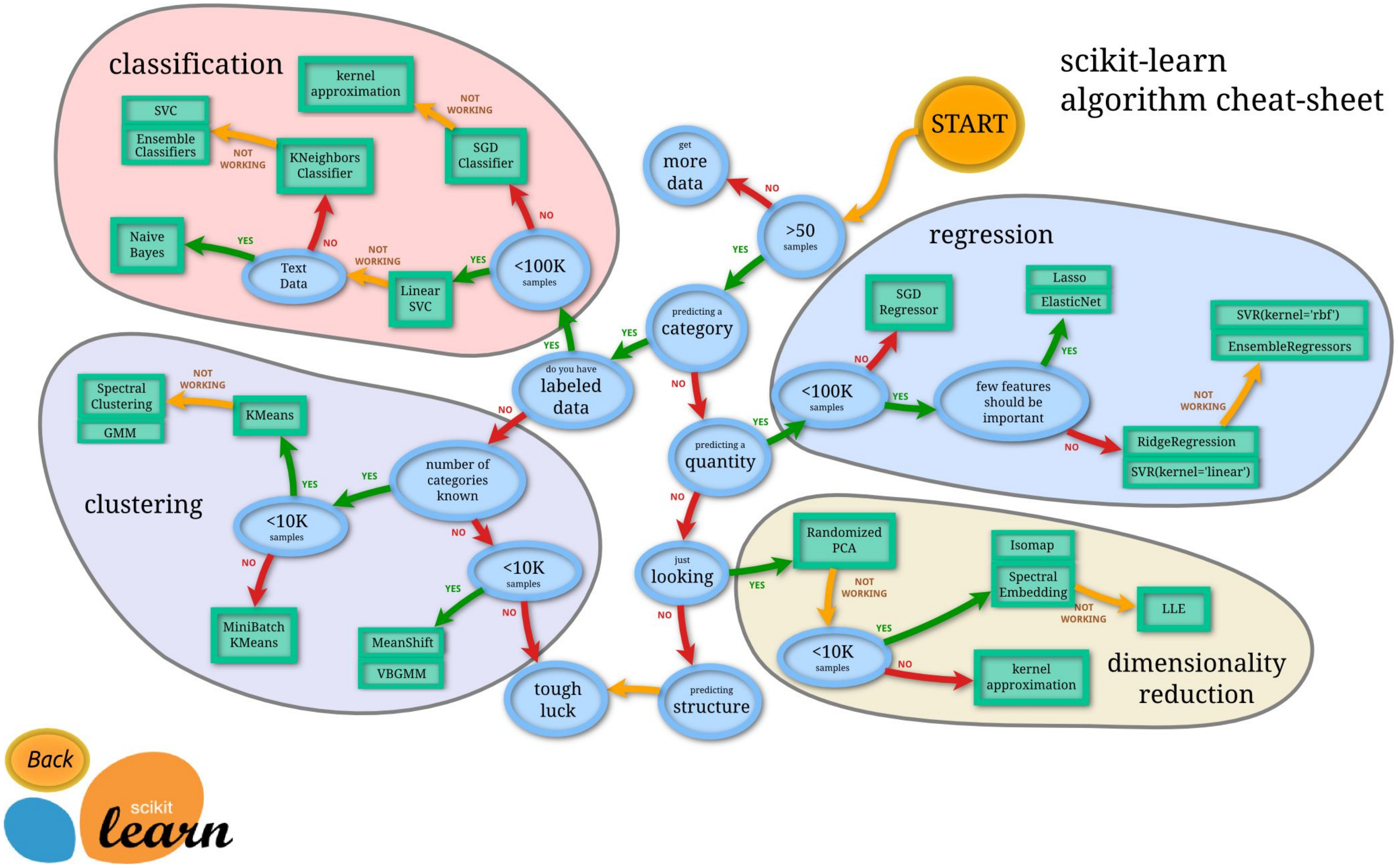
$$x^{(1)} = \langle \alpha_1^{(2)}, \alpha_2^{(2)}, \ldots, \alpha_{20}^{(2)} \rangle$$

What dimension are we using now?

192×198 (30000 pixels) ⇒ 20

# Model Zoo

Lots of models!



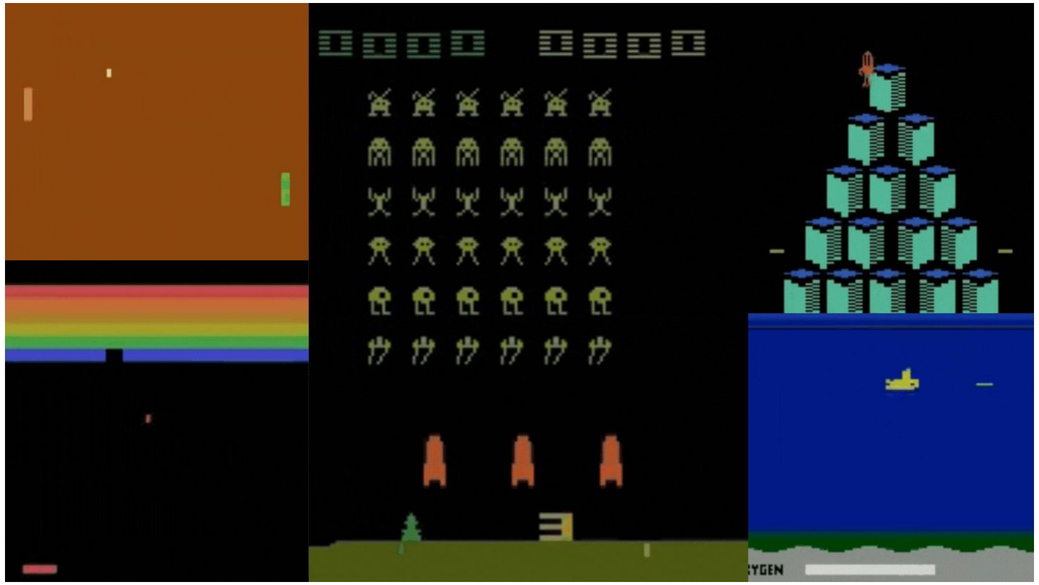
# Q&A Break

# Outline

- **Review from last time**
  - Supervised, Unsupervised, Reinforcement Learning
- **Supervised Learning concepts**
  - Features, models, training, other terminology
- **Unsupervised Learning concepts**
  - Clustering, dimensionality reduction
- **Reinforcement Learning concepts**
  - Credit-assignment, Exploration vs. Exploitation

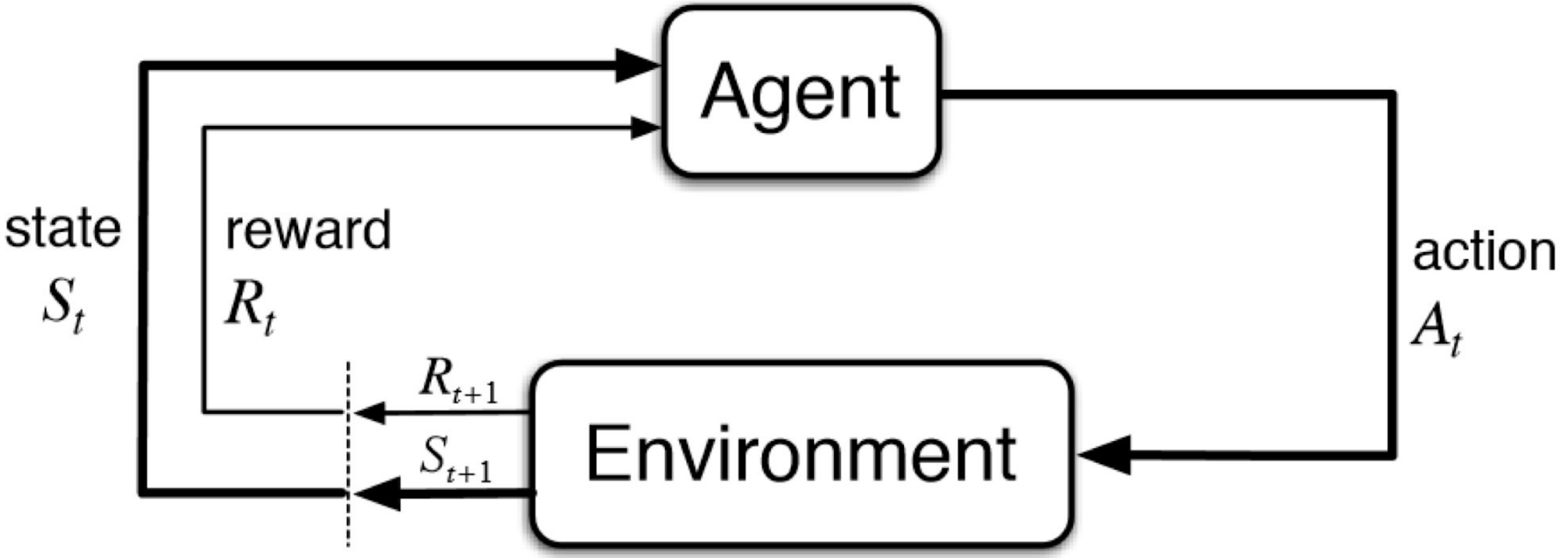
# Reinforcement Learning

- Given: an agent that can take actions at some environment states and a reward function specifying how good an action is.
- **Goal:** learn to choose actions that maximize future reward total.



# Reinforcement Learning

- Given: an agent that can take actions at some environment states and a reward function specifying how good an action is.
- **Goal:** learn to choose actions that maximize future reward total.



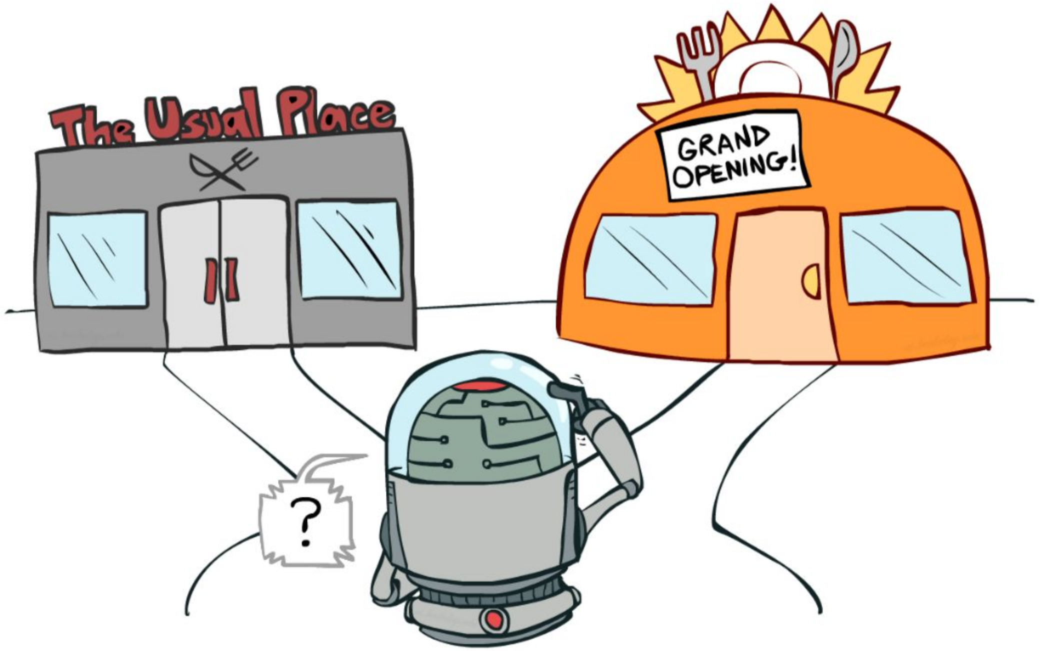
Agent collects data $s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_T, a_T, r_T$ .

Learn *policy* $\pi : \mathcal{S} \rightarrow \mathcal{A}$ that maximizes $\sum_{t=0}^{\infty} \gamma^t r_t$.

# **Reinforcement Learning**: Key Problems

1. Problem: actions may have delayed effects.
   - Requires **Credit-assignment**
2. Problem: maximal reward action is unknown
   - **Exploration-exploitation trade-off**

“..the problem [exploration-exploitation] was proposed [by British scientist] to be dropped over Germany so that German scientists could also waste their time on it.”
- Peter Whittle



# Q&A Break

# Thanks Everyone!

Some of the slides in these lectures have been adapted/borrowed from materials developed by Mark Craven, David Page, Jude Shavlik, Tom Mitchell, Nina Balcan, Elad Hazan, Tom Dietterich, Pedro Domingos, Jerry Zhu, Yingyu Liang, Volodymyr Kuleshov, Fred Sala, Kirthi Kandasamy, Josiah Hanna, Tengyang Xie