

EECS 545: Machine Learning

Lecture 1. Introduction

Honglak Lee

1/10/2024



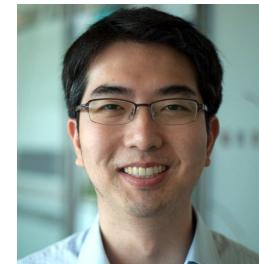
Outline

- Administrative
- What is machine learning?

Teaching staff

Instructors:

- Honglak Lee (honglak@umich.edu, Office: CSE 3773)



Graduate Student Instructors:

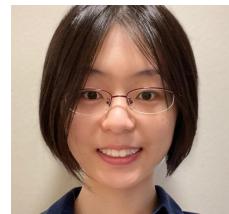
- Yunseok Jang (yunseokj@umich.edu)
- Anthony Liu (anthliu@umich.edu)
- Violet Fu (violetfy@umich.edu)



Yunseok



Anthony



Violet

Please contact us via [Piazza](#) or eeecs545.staff@gmail.com (please avoid sending to personal emails unless it's really necessary)

- For office hours, please see future announcements and [calendar](#)

About this course

- Graduate-level introduction of machine learning
- Foundations of machine learning
 - mathematical derivation, implementation of the algorithms, applications
- Topics
 - supervised learning
 - unsupervised learning
 - reinforcement learning
- Other topics
 - deep learning, learning theory, probabilistic models, sparsity and feature selection, Bayesian techniques, ensemble methods

About this course

- Cover practical applications of machine learning
 - computer vision, speech recognition, language understanding, data mining, robot perception and control, etc.
- Our goal is to help you to
 - understand fundamentals of machine learning
 - learn technical details of ML algorithms
 - learn how to implement some important algorithms
 - use machine learning algorithms for your research and applications.

Textbooks

There will be no official textbook for the course. However, the following materials will be helpful (all of them are available online):

- Chris Bishop, “[Pattern Recognition and Machine Learning](#)”. Springer, 2007.
- Kevin Murphy, “Machine Learning: A Probabilistic Perspective”, 2012.
- David Barber, “[Bayesian Reasoning and Machine Learning](#)”, 2017.
- Ian Goodfellow and Yoshua Bengio and Aaron Courville, “[Deep Learning](#)”,
- Hastie, Tibshirani, Fiedman, “[Elements of Statistical Learning](#)”. Springer, 2010.
- Sutton and Barto, "[Reinforcement Learning: An Introduction](#)," MIT Press, 2018.
- (optional) Boyd and Vandenberghe, "[Convex Optimization](#)," Cambridge University Press, 2004.
- (optional) Mackay, “[Information Theory, Inference, and Learning Algorithms](#)”. Cambridge University Press. 2003.

Prerequisites

- Undergrad-level linear algebra (e.g., MATH 217, MATH 417)
OR graduate matrix methods courses (EECS 505 / EECS 551) which have several relevant linear algebra concepts
- Multivariate calculus
- Undergrad probability and statistics (e.g., EECS 301)
- Programming skills (equivalent to EECS 280, EECS 281, and experience in Python)
 - non-trivial level of programming is required.
- NOTE: **If you have not taken all of the above (or don't have equivalent background knowledge), seriously reconsider taking EECS 545 (otherwise, your workload will be extremely high).** We strongly recommend that you finish them first before taking this course.

Grading policy

- Homework: 45%
- Quiz: 5%
- Exam: 25% (tentative date: April 10)
 - [Honor code](#)
- Project: 25%
 - Progress report (5%) Due: Mar 12
 - Poster presentation (5%) Date: Apr 26
 - Final project report (15%) Due: May 1
- Extra credits: Up to 3% may be awarded for active participation in class and piazza (details TBD).
 - Note: anonymous posts/comments are not counted

Reference: Statistics from EECS 545 WN 2022 and WN 2023

Aggregate grade distribution

Grade	Percentage
A+	2%
A	31%
A-	26%
B+	22%
B	11%
B-	2%
C+	1%
D	2%
F	4%

Average performance statistics

Grade	Total points (%)	HW (%)	Exam (%)	Project (%)
A+	96	100	93	90
A	89	99	77	83
A-	84	96	67	80
B+	79	92	61	77
B	74	81	60	75
B-	68	70	50	79
C+	53	37	49	79
Global Avg.	83	93	68	80

* Note: this is smoothed (aggregated) version of statistics from the two semesters and is provided only for illustration purpose. W24 grade distribution may be subject to change from the above.

Language of Choice: Python

- **Python** is a great language overall for machine learning, with huge ecosystem and many modern libraries for various tasks
- We will use **Python 3 (3.11+)** throughout the course.
 - We will be using popular scientific computing libraries, such as **numpy**, **matplotlib**, and **pytorch**.
- There will be a tutorial session video for basics of python and numpy in Canvas by the end of this week.

Homework

- There will be 5 problem sets.
 - **The best 4 out of 5 scores will be used for final grading.**
- Goal: strengthen the understanding of the fundamental concepts, mathematical formulations, algorithms, detailed implementations, and the applications.
- The problem sets will also include **programming assignments** to implement algorithms covered in the class.
- A typeset document (e.g. LaTeX) is recommended, to avoid misunderstandings due to poor handwriting when grading.
- Homework #1 will be out by next Tuesday (1/16):
Due 1/30, 11:55 PM via [Gradescope/Autograder](#) (roaster will be synced every 3 days).

Late days

- No homework will be accepted 3 days after the due date.
- Total 9 late days allowed:
 - Up to 3 maximum late days can be used per assignment.
 - After using up all late days, your assignment will be penalized by 20% from your scores (which can significantly hurt your grade).

Study group

- Form your study group early on!
 - Up to five people are allowed.
- For homework, you may discuss between the study group members, but you **should** submit your own solution *independently*.
- In your homework submissions, you must put:
 - the names of other people you collaborated with (if not, your homework will be flagged as **plagiarism**).
 - Submission time.
- Please start on homework early.
(Warning: cramming will not work!)

Quiz

- We will upload a Quiz online (approximately for each lecture).
- You are allowed to take quiz up to 3 times until the next day of each lecture (*24 hour window*).
- We will announce the details (the time/format) as soon as ready.

Honor Code

- No cheating of any kind shall be made.
- Previously, several incidents of violation of honor (plagiarism in homework submissions and cheatings in exams) were reported.
 - You will get into a **big trouble** if you violate the honor code. You will fail the course.
- Please review the UofM [Honor Code](#).

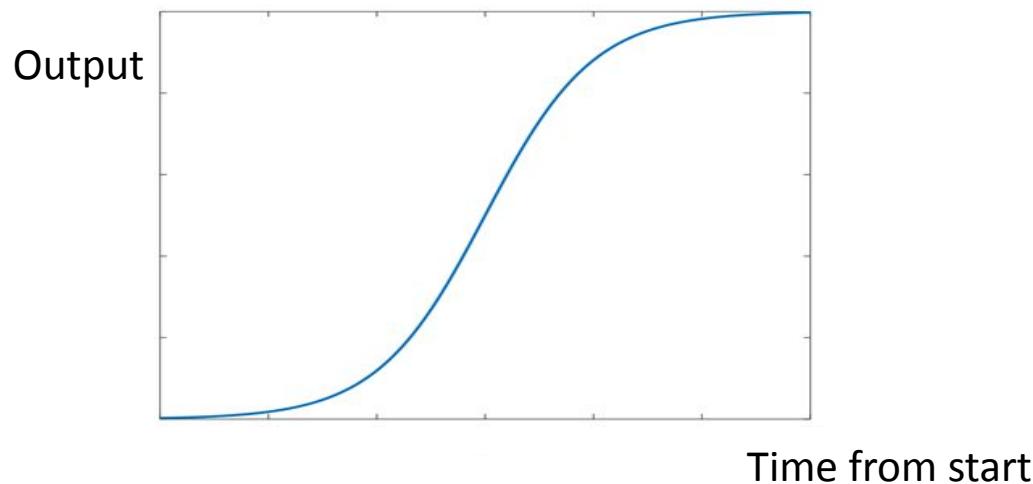
Course Project

- Scope
 - Develop and apply algorithms in machine learning for solving real-world problems
 - We will provide a predefined list of project topics (TBD)
- Milestones (tentative)
 - Project proposal due: **February 2, 2024**
 - Project progress report due: **March 12, 2024**
 - Final project poster presentation: **April 26, 2024 (Friday)**
 - Final project report due: **May 1, 2024 (No Late Days)**
- Requirements
 - Write a 8-page paper
 - Submit the final code
 - Give a poster presentation
- Evaluation is based on:
 - Novelty, technical quality, significance, and presentation quality of the project.

Note: More information will be provided soon (TBD).

Course Project

- 4 or 5 people can form a project group.
- Talk to instructor if you want to get suggestions about project topics.
- Start early! (form your group and start working)



Course Calendar ([link](#))

EECS 545 W24

Today January 2024 ▾

Print Week Month Agenda ▾

Mon	Tue	Wed	Thu	Fri	Sat	Sun
Jan 1		2	3	4	5	6
8	9	10	11	12	13	14
		Classes begin Project suggestion form release 9am 545 Lecture				
15	16	17	18	19	20	21
MLK day	HW #1 release 9am 545 Lecture					
22	23	24	25	26	27	28
9am 545 Lecture		9am 545 Lecture				
29	30	31	Feb 1	2	3	4
9am 545 Lecture	[info] Add/Drop deadline HW #1 due HW #2 release 9am 545 Lecture		project proposal due (tentative)			

Events shown in time zone: Eastern Time - New York

+ Google Calendar

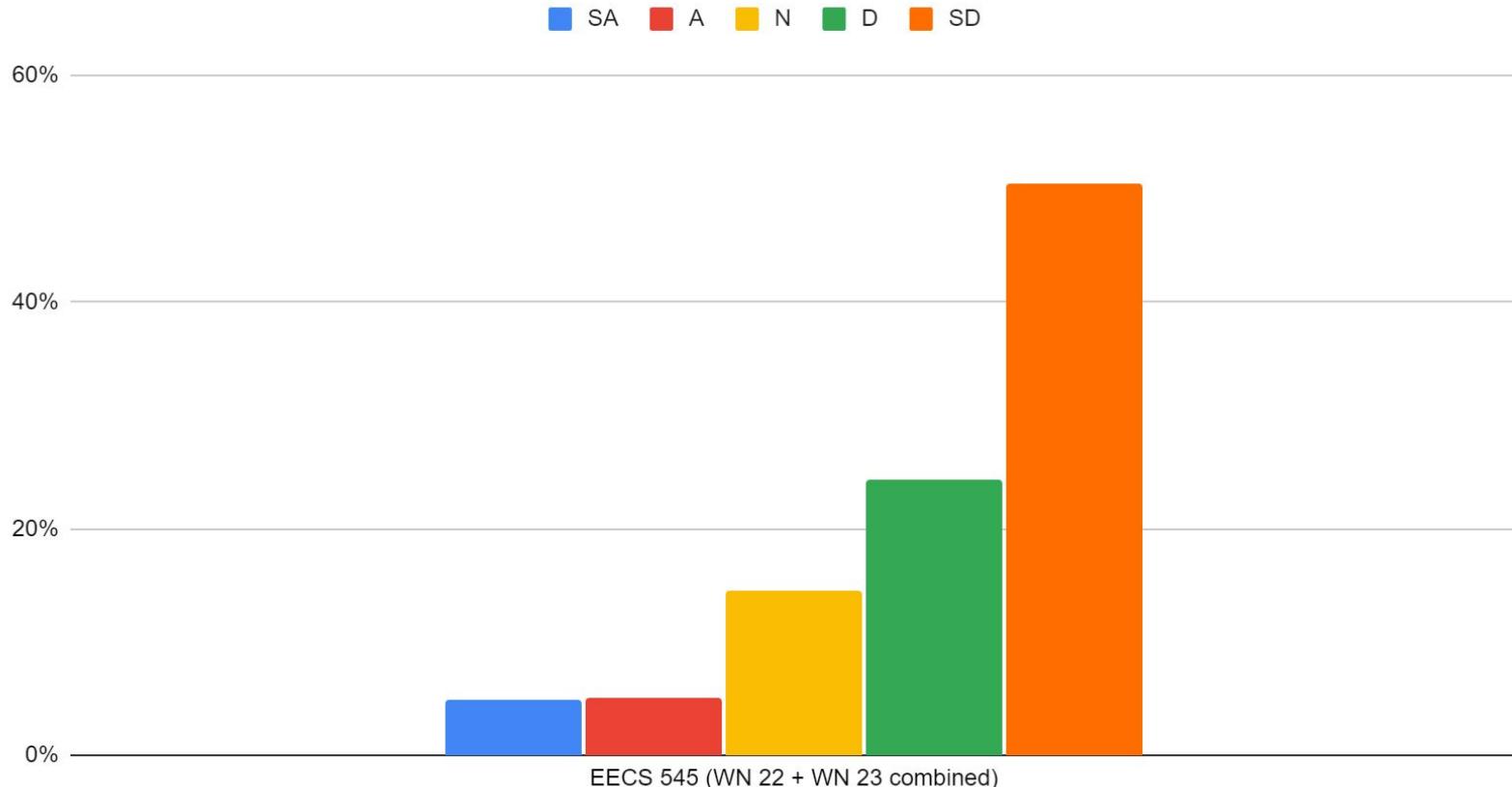
(Tip: you can add EECS 545 calendar to your personal google calendar by clicking "+ Google Calendar" in the bottom right corner.)

Expect a high workload! (*Warning*)

- While EECS 545 is a 3 unit course, the **expected workload is comparable to “heavy” 4 unit courses.**
 - If you really need to get 4 credits, EECS 545 may not be the right course for you.
- Previously, students who took EECS 545 (WN21+WN22) reported the relative workload as “lighter” (~5%), “typical” (~15%), “heavier” (~24%), or “much heavier” (~50%).
- You can expect to learn a lot of things but the workload could feel **extremely heavy** if you don’t have sufficient background in math (e.g., Linear algebra, Calculus, Probabilities) and/or programming (e.g., nontrivial programming background).

Expect a high workload! (*Warning*)

As compared with other courses of equal credit, the workload for this course was (SA=Much Lighter, A=Lighter, N=Typical, D=Heavier, SD=Much Heavier). (Q891)



Other Information

- Review sessions
 - Will hold review sessions on background materials (Linear Algebra, Probability, Python).
 - Dates: TBD
- Exam
 - Tentatively scheduled for April 10, 2024
- Beginning-of-course survey :
<https://forms.gle/XBgCPRuyxAtxW1yv5> (open though Jan 12)

Any questions?

Outline

- Administrative
- What is Machine Learning?

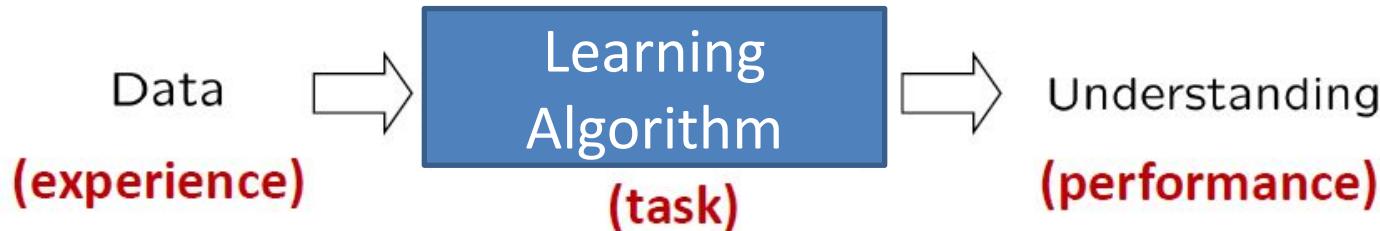
Definition of Machine Learning

Formal definition (Mitchell 1997):

- 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.

Informal definition

- Algorithms that improve their prediction performance at some task with experience (or data).



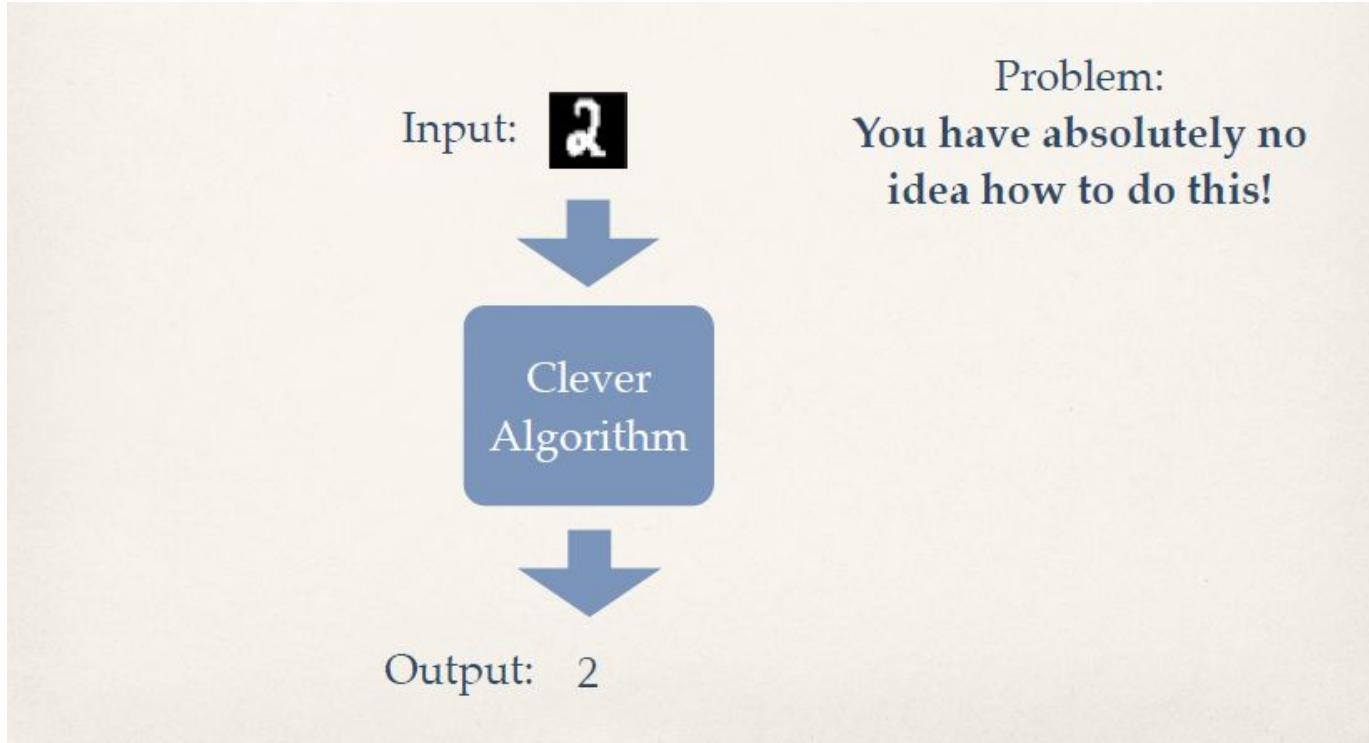
Example: Spam email filtering

“A computer program is said to *learn* from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.”

- Suppose your email program watches which emails you do or do not mark as spam, and based on that learns how to better filter spam.
- Task:
 - Classifying emails as spam or not spam.
- Experience:
 - Watching you label emails as spam or not spam.
- Performance measure
 - The number (or fraction) of emails (disjoint from the training emails) correctly classified as spam/not spam.

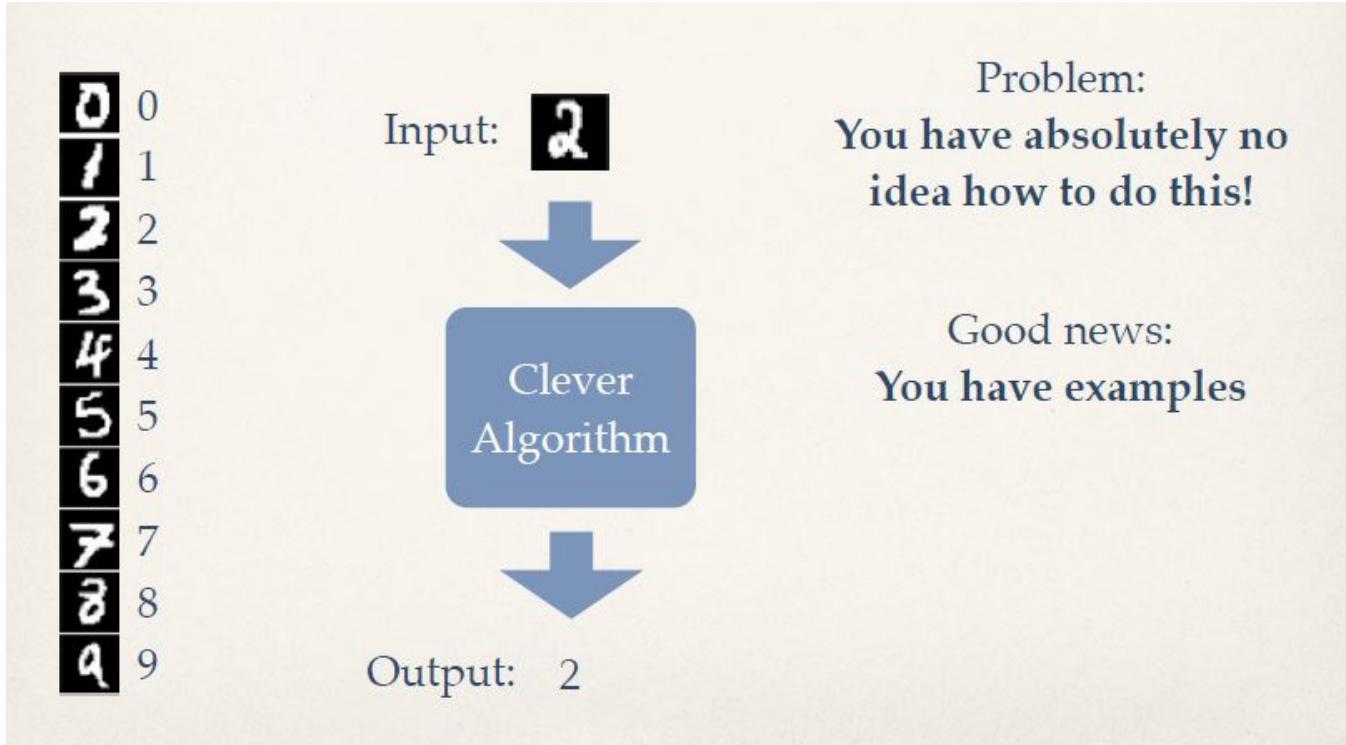
Example

- Problem: Given an image of a handwritten digit, what digit is it?



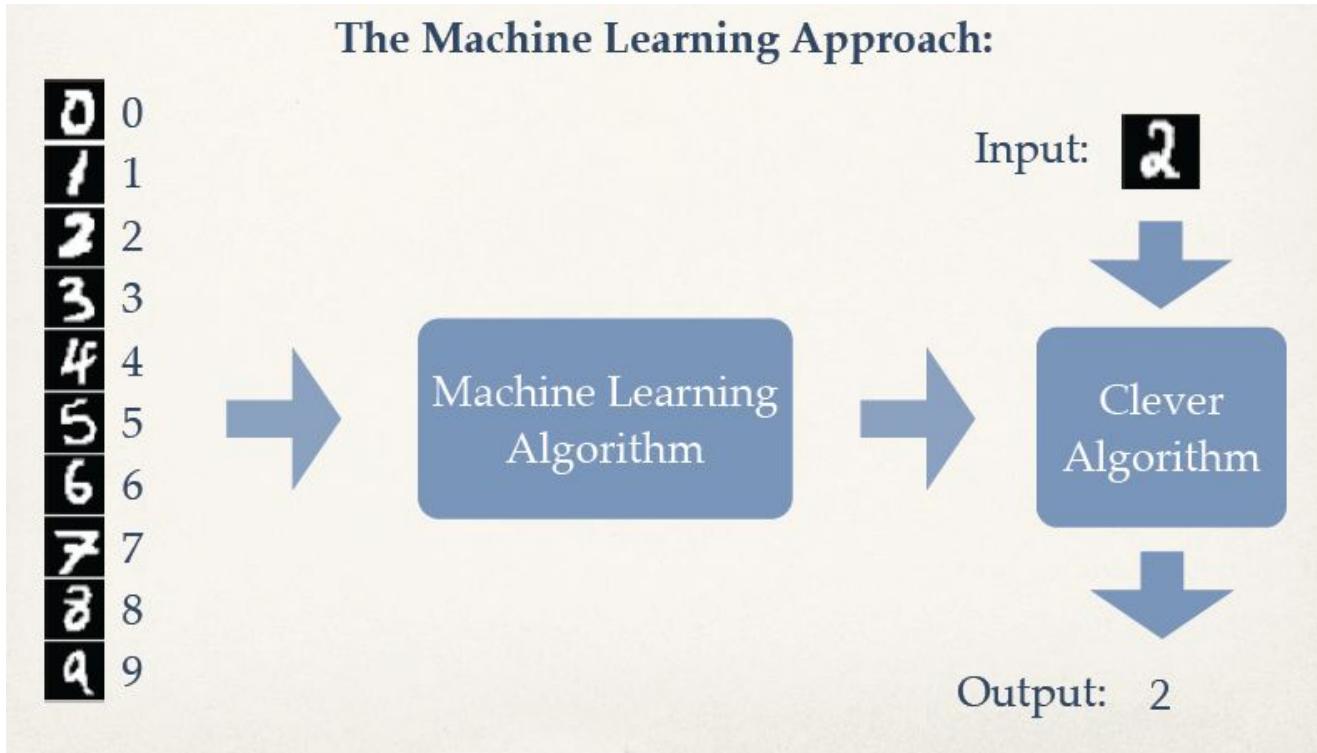
Example

- Problem: Given an image of a handwritten digit, what digit is it?



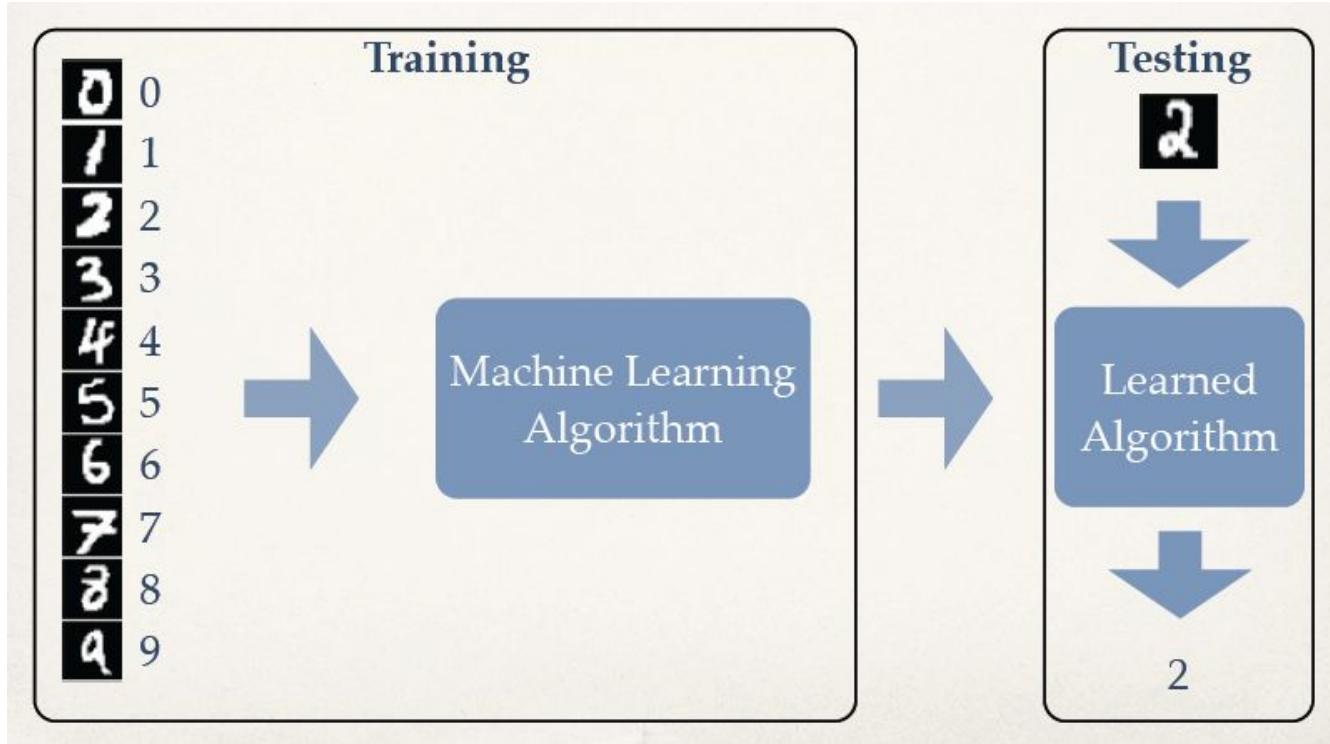
Example

- Problem: Given an image of a handwritten digit, what digit is it?



Example

- Problem: Given an image of a handwritten digit, what digit is it?



Machine Learning Tasks

- Supervised Learning
 - Classification
 - Regression
- Unsupervised Learning
 - Clustering
 - Density estimation
 - Embedding / Dimensionality reduction
- Reinforcement Learning
 - Learning to act (e.g., robot control, decision making, etc.)

Supervised Learning

Given a dataset $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$,
where

- $x_i \in \mathcal{X}$: input (feature)
- $y_i \in \mathcal{Y}$: output (label)

a black box ML algorithm produces a
prediction function $h : \mathcal{X} \rightarrow \mathcal{Y}$, such that
 $h(x)$ can predict the y values for all x
(including training data $x_i \in D$ and unseen
test data x^*).

Supervised Learning

- Labels could be discrete or continuous
 - Discrete labels: **classification**
 - Continuous labels: **regression**

Supervised Learning - Classification

Feature Space \mathcal{X}



Words in a document

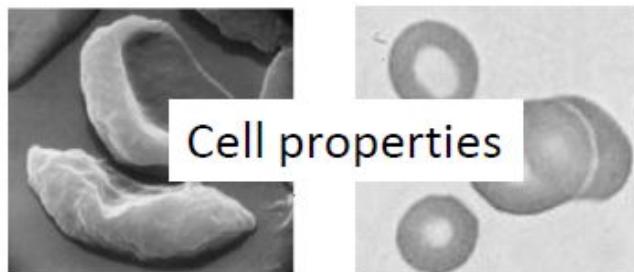
Label Space \mathcal{Y}

- “Sports”
- “News”
- “Science”

...



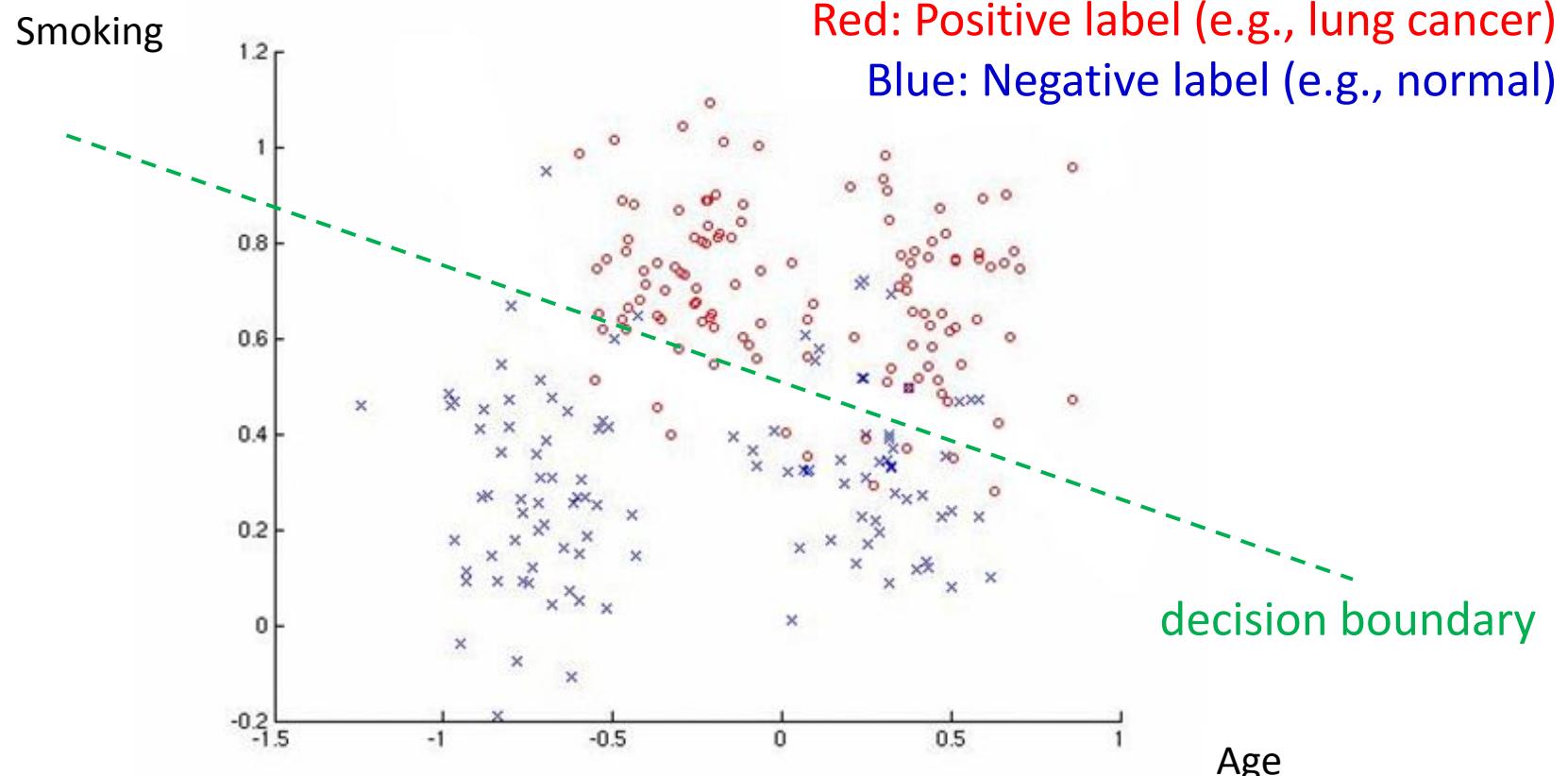
Cell properties



- “Anemic cell”
- “Healthy cell”

Discrete Labels

Supervised Learning - Classification



“Learning decision boundaries”

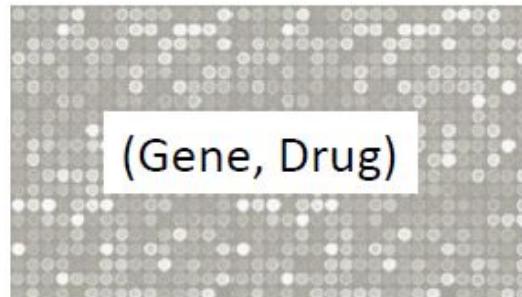
Supervised Learning - Regression

Feature Space \mathcal{X}



Label Space \mathcal{Y}

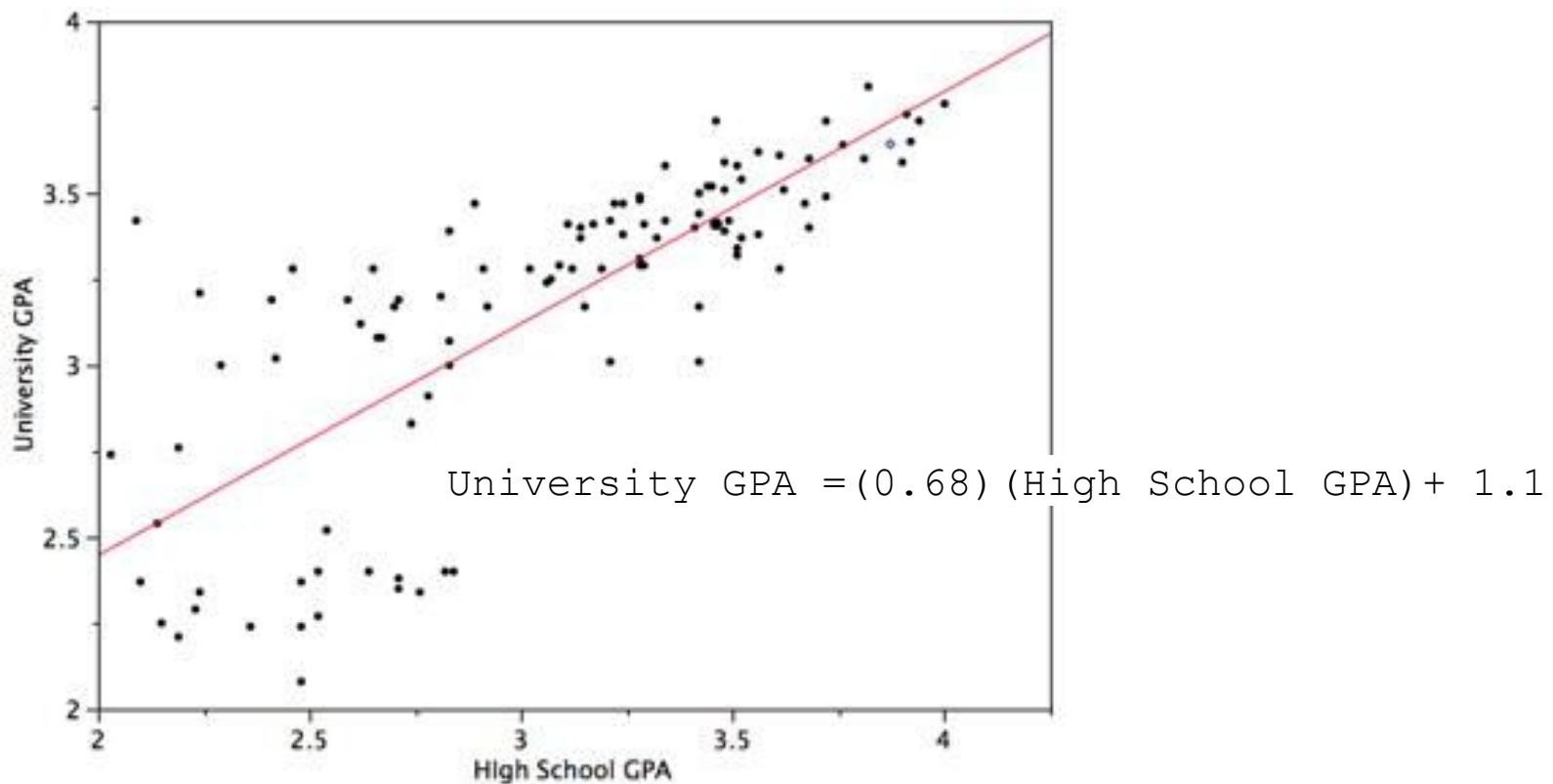
Share Price
“\$ 24.50”



Expression level
“0.01”

Continuous Labels

Supervised Learning - Regression



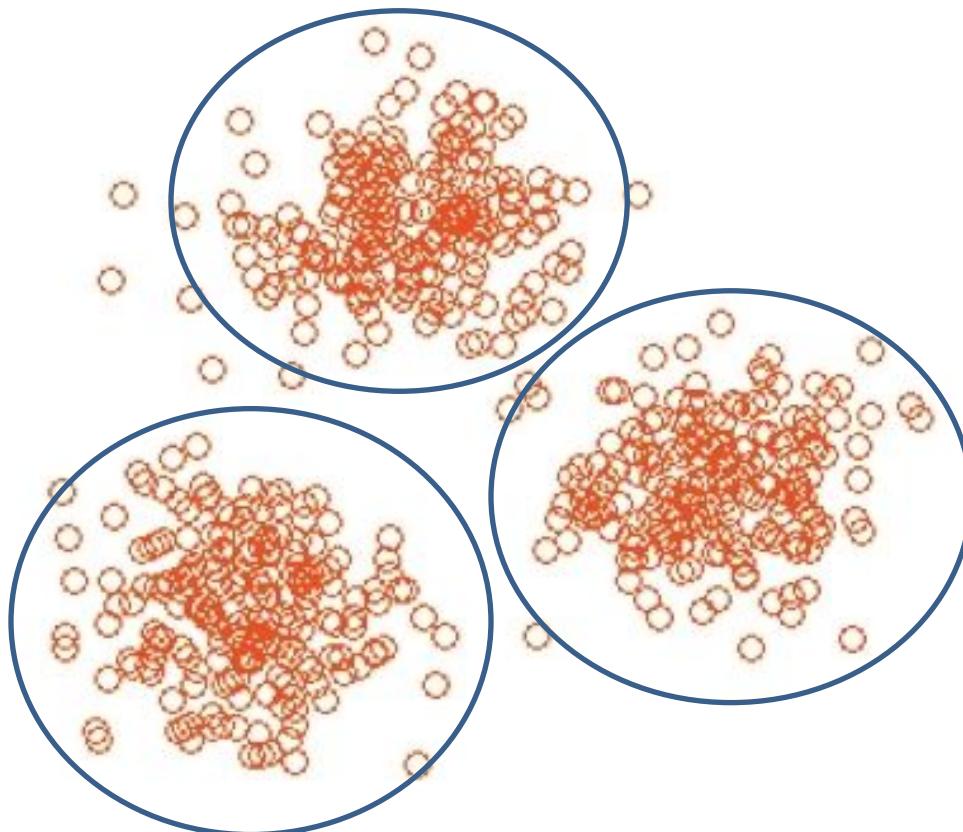
“Learning regression function $f(X)$ ”

Unsupervised Learning

- Goal:
 - Given data X without any labels
 - Learn the **structures** or **distribution** of the data
 - Clustering
 - Probability distribution (density)
 - Generating data
 - Embedding & neighborhood relations
- “Learning without teacher (supervision)”

Unsupervised Learning – Clustering

- “Grouping into similar examples”



Unsupervised Learning – Clustering

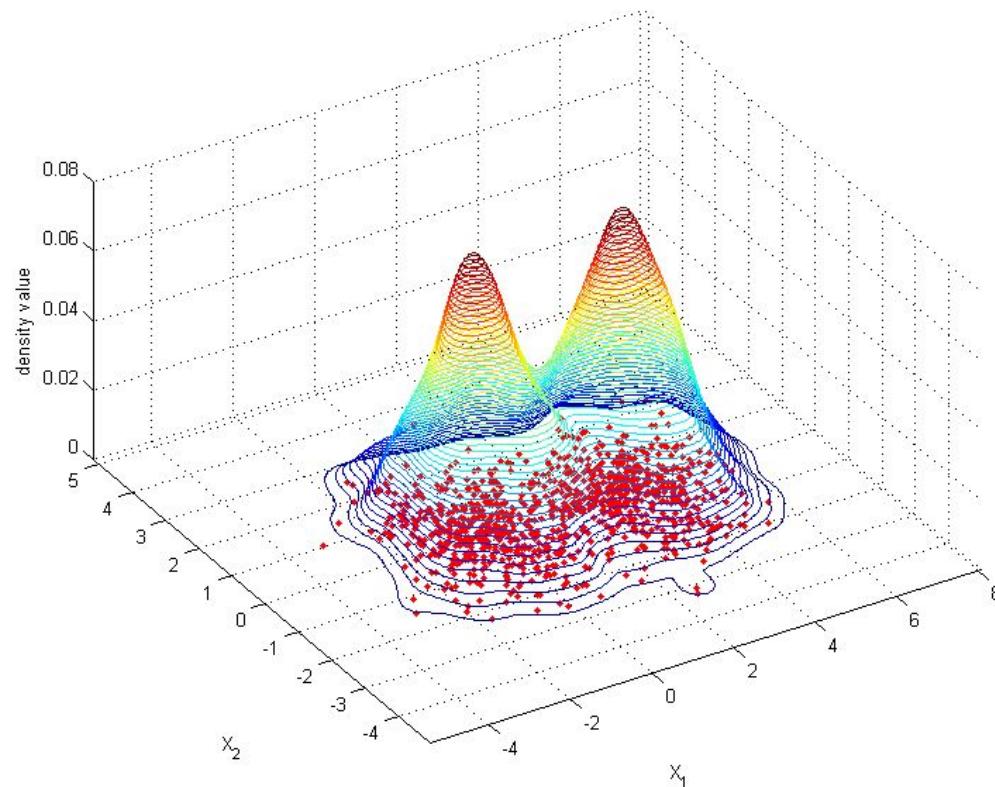
Group similar things e.g. images

[Goldberger et al.]



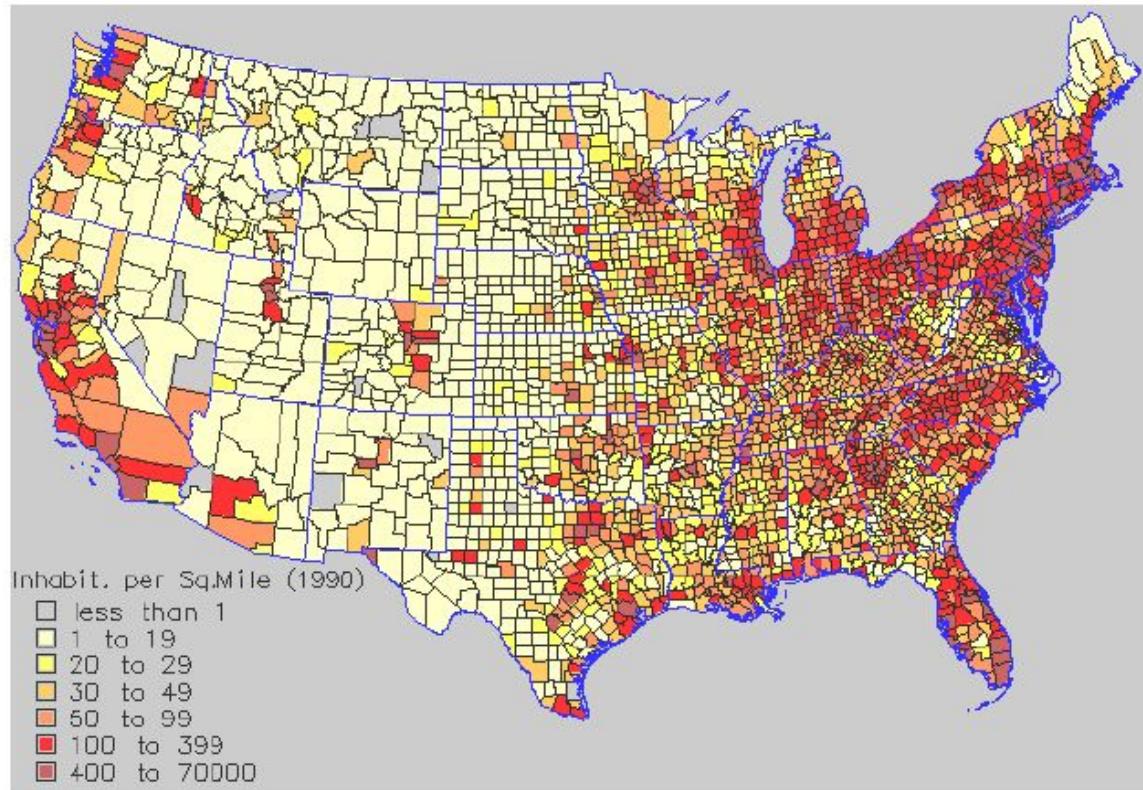
Unsupervised Learning – Density estimation

$P(X_1, X_2)$
“Probability”



Unsupervised Learning – Density estimation

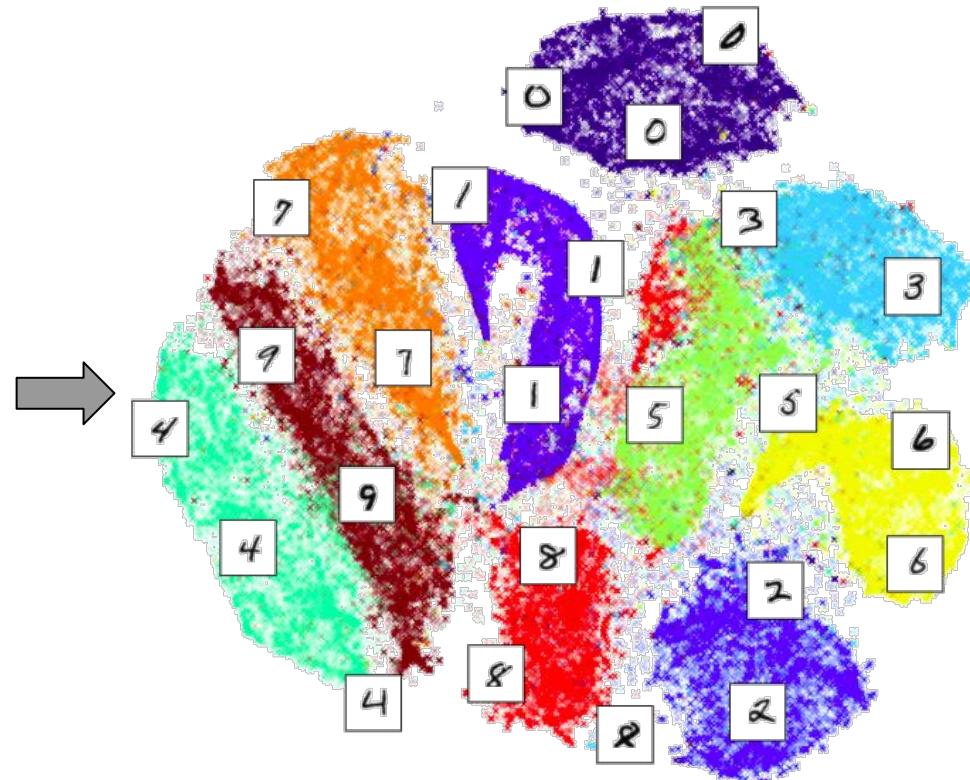
Population density



Slide credit: Aarti Singh

Unsupervised Learning-Embedding and Dimensionality reduction

- E.g., Reducing handwritten digits (784 dim) into low dimensional coordinates



[Maaten and Hinton, 08]

Reinforcement Learning

- Setting
 - Given sequence of states s_t and “rewards” r_t (e.g., delayed labels)
 - Agent has to take actions a_t for each time step t
- Goal:
 - How to “learn to act” or “make decisions” to maximize the sum of future rewards
- Example: Robot navigation task
 - Input: Dynamical environment + sensor input
 - Action: control signals
 - Rewards: time to reach goal without colliding with obstacles

Reinforcement Learning – learning to control

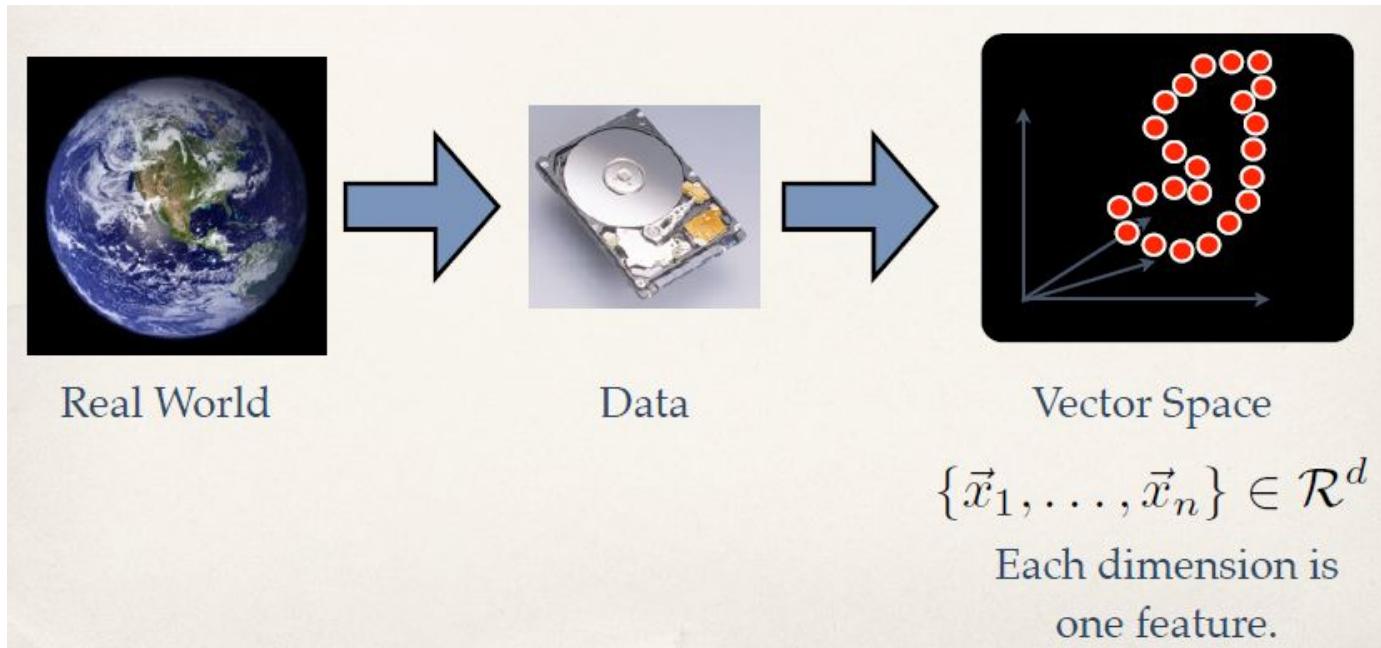
- Example: Robot walking
 - States: sensor inputs, joint angles
 - Action: servo commands for joints
 - Rewards:
 - 1 for reaching the goal
 - -1 for falling down
 - 0 otherwise
- Goal: How can we provide control inputs to maximize the expected future rewards?



Feature representations

Feature Extraction

- Represent data in terms of vectors.
 - Features are **statistics** or **attributes** that describe the data.



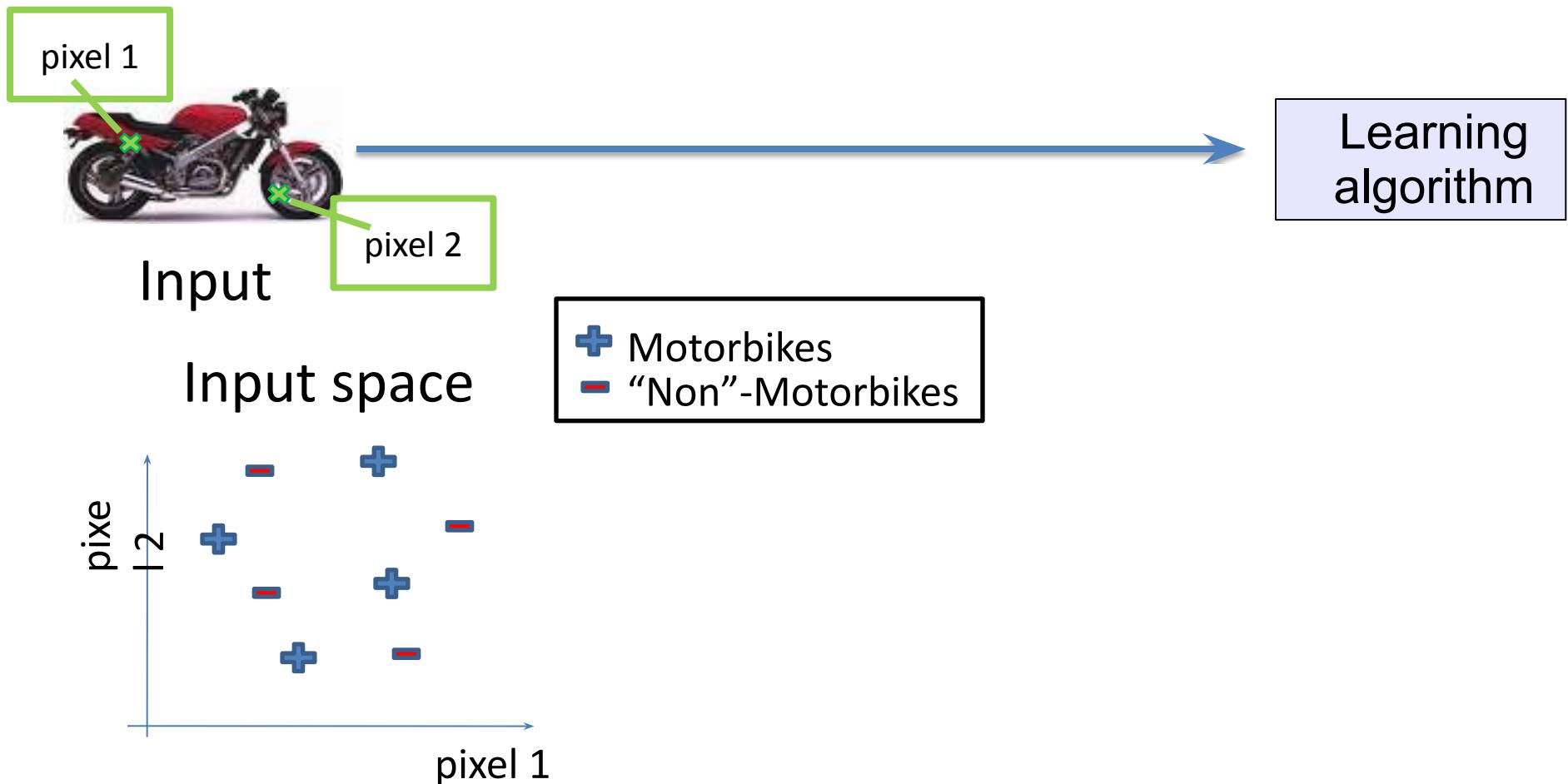
Examples of features: Housing data

- **Given statistics about houses in a local area, predict median value of homes.**
 - #ROOM: average number of rooms per dwelling
 - AREA: average area of house in square foot
 - AGE: proportion of owner-occupied units built prior to 1940
 - CRIME: per capita crime rate by town
 - RESZONE: proportion of residential land zoned for lots over 25,000 sq.ft.
 - INDUS: proportion of non-retail business acres per town
 - NOX: nitric oxides concentration (parts per 10 million)
 -
- **Label: Median value of owner-occupied homes**

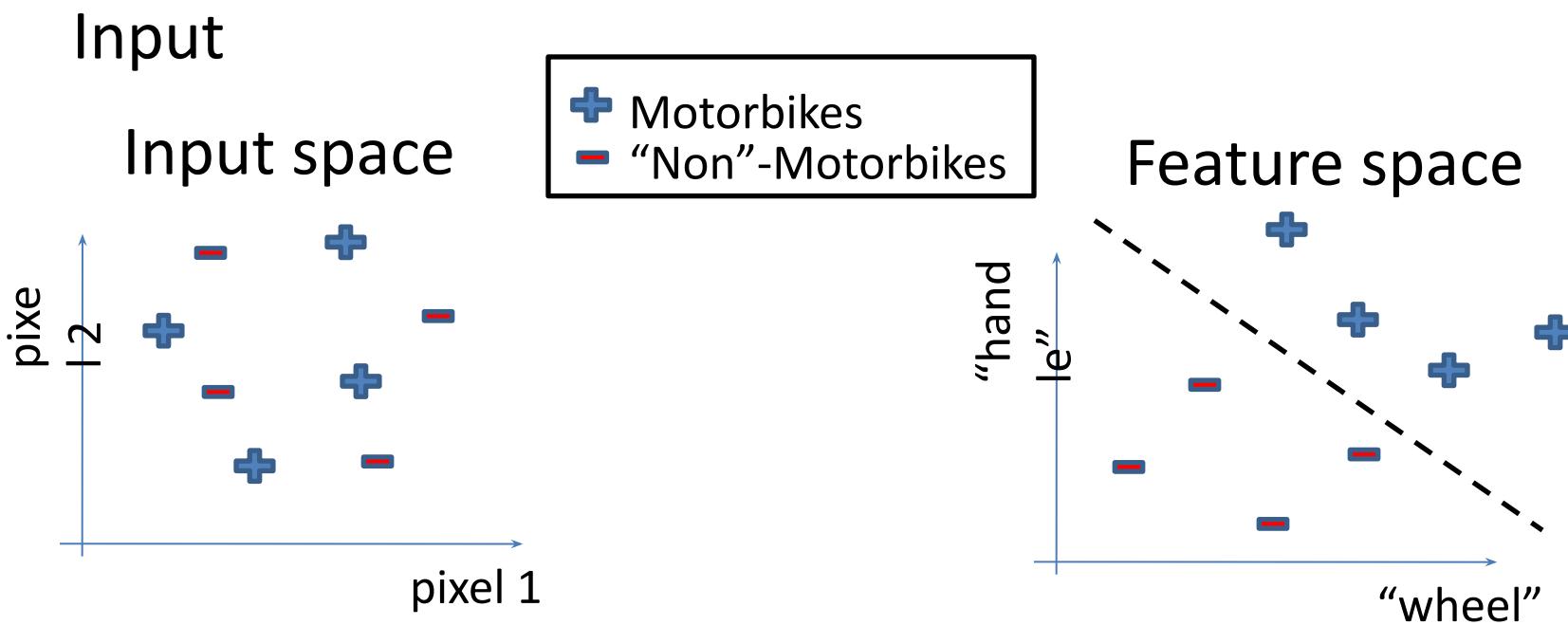
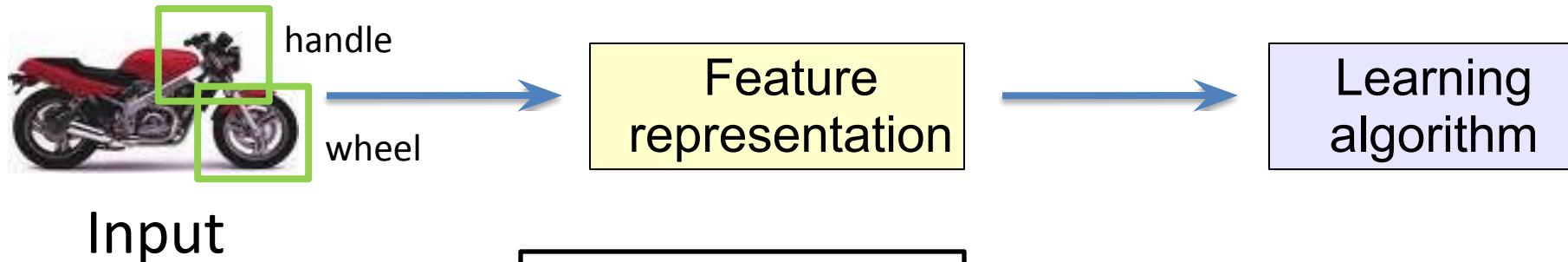
Examples of features: Recognizing handwritten-digits

- Input: 28×28 pixel images
- Output: Class Labels $\in \{0, 1, 2, \dots, 9\}$
- The following basic features can be used:
 - Pixel Values (784 dimensional vectors)
 - Aspect Ratio of the tight bounding boxes
 - Existence of long vertical strokes
 - Existence of long horizontal strokes

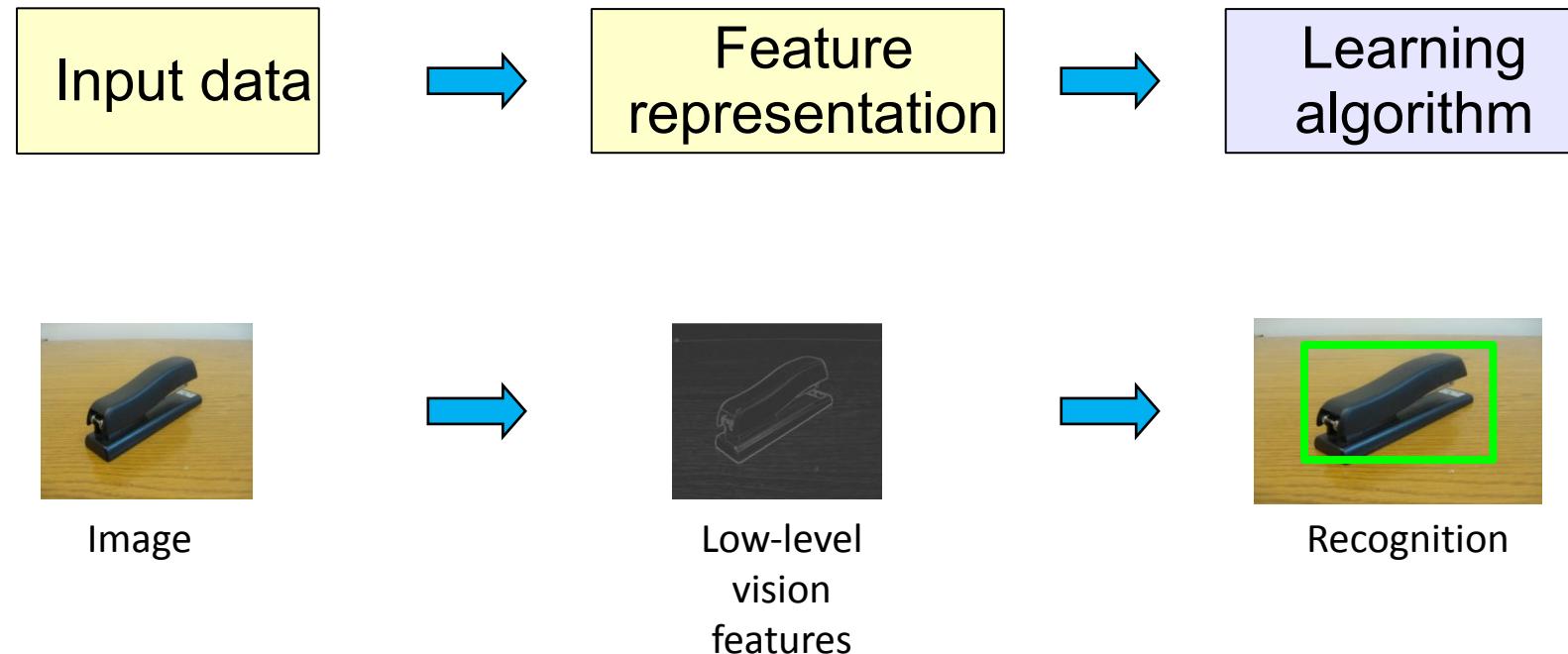
Learning pipeline



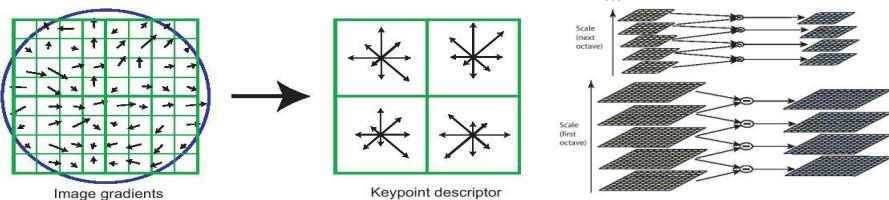
Learning pipeline



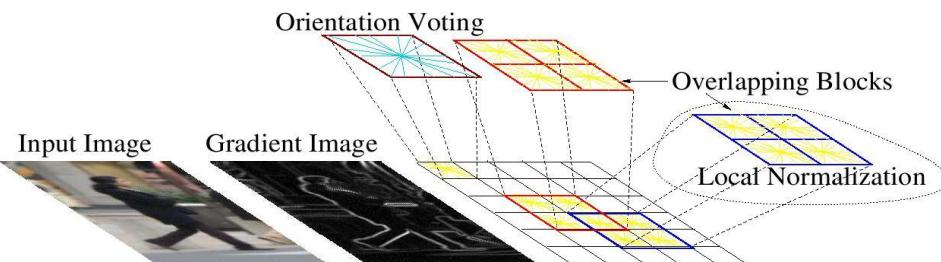
(Traditional) Computer Perception Pipeline



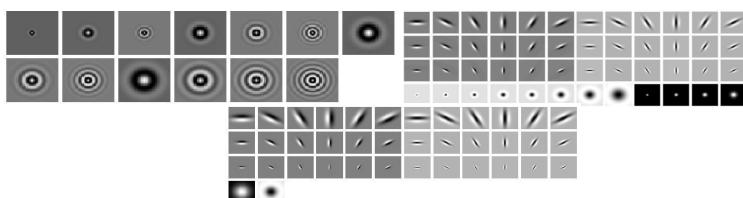
(Traditional) Computer vision features



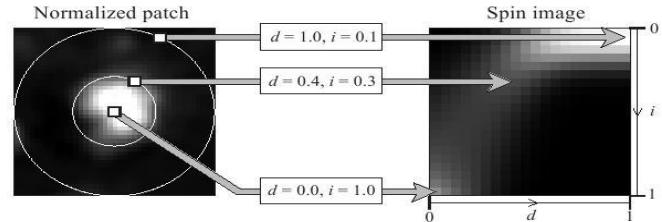
SIFT



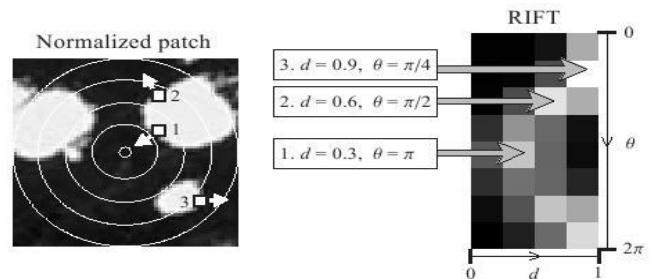
HoG



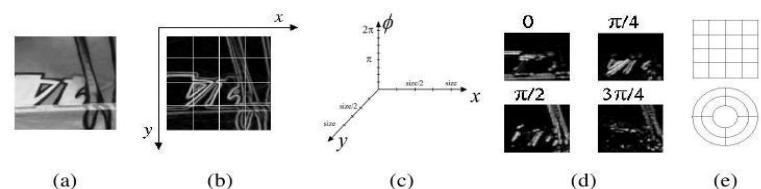
Textons



Spin image



RIFT



GLOH

Learning feature hierarchies with Deep Learning

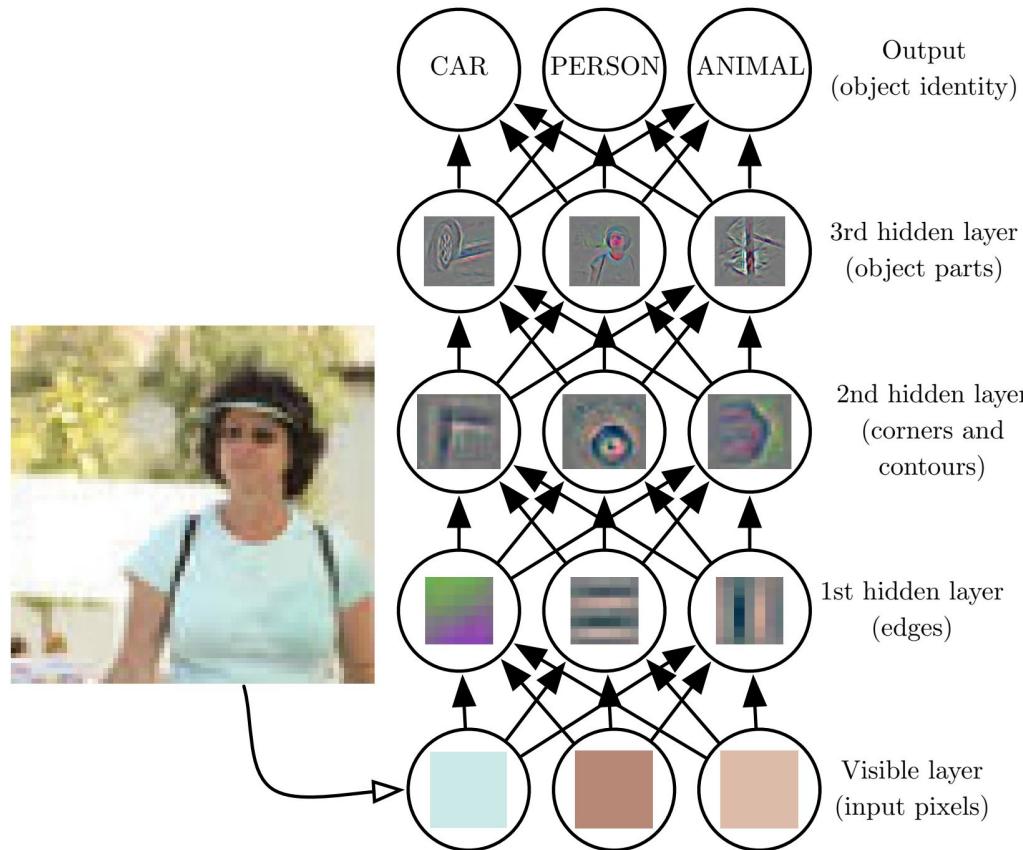


Figure source: Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep Learning, 2016

ML applications

Examples of ML applications

- Computer vision
- Speech recognition
- Natural language processing
- Robotics
- Medical image recognition
- Time series prediction/classification
-

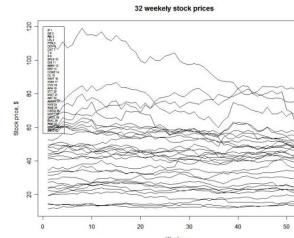
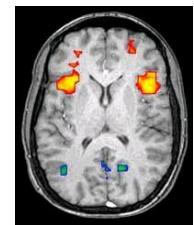
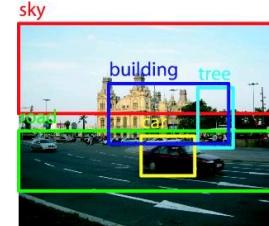


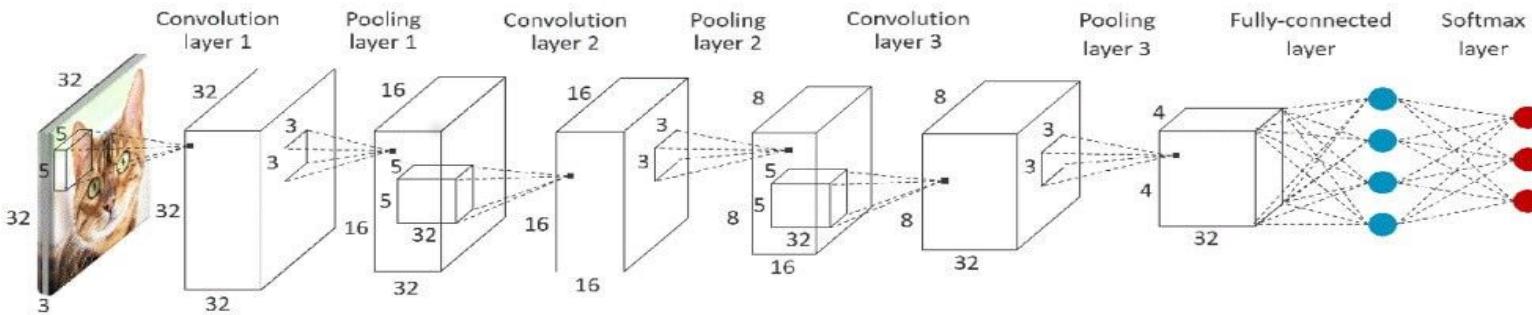
Image classification

- Goal: predict the class of input image

Training dataset
(image, label)



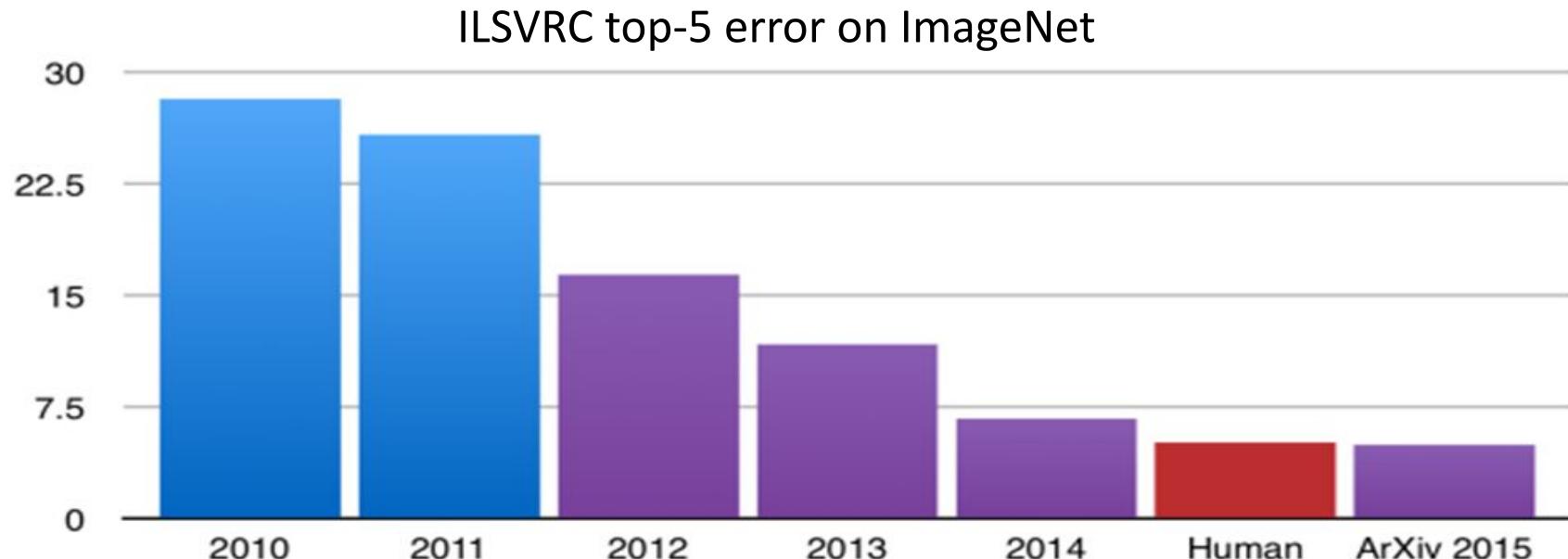
Input



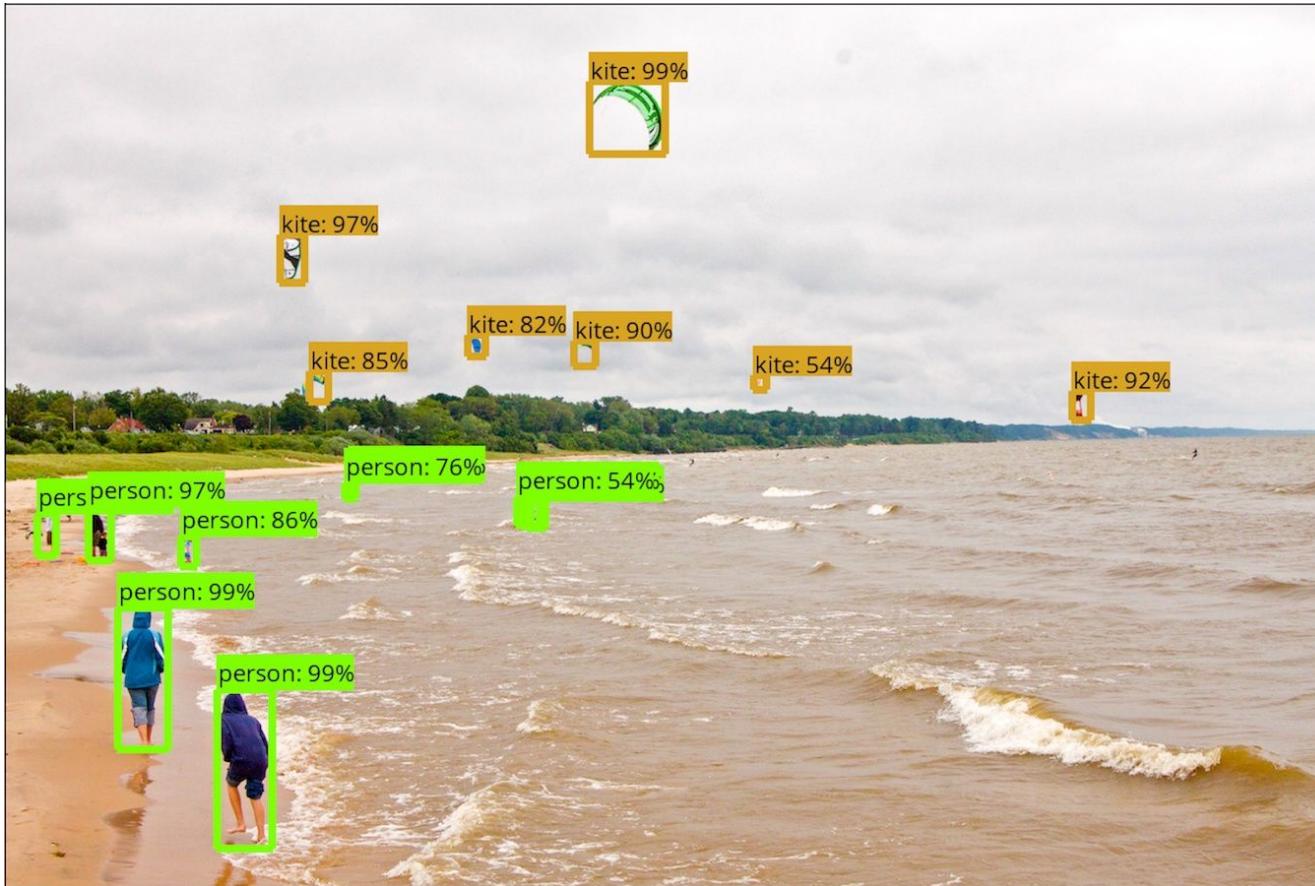
Prediction

cat

Image classification

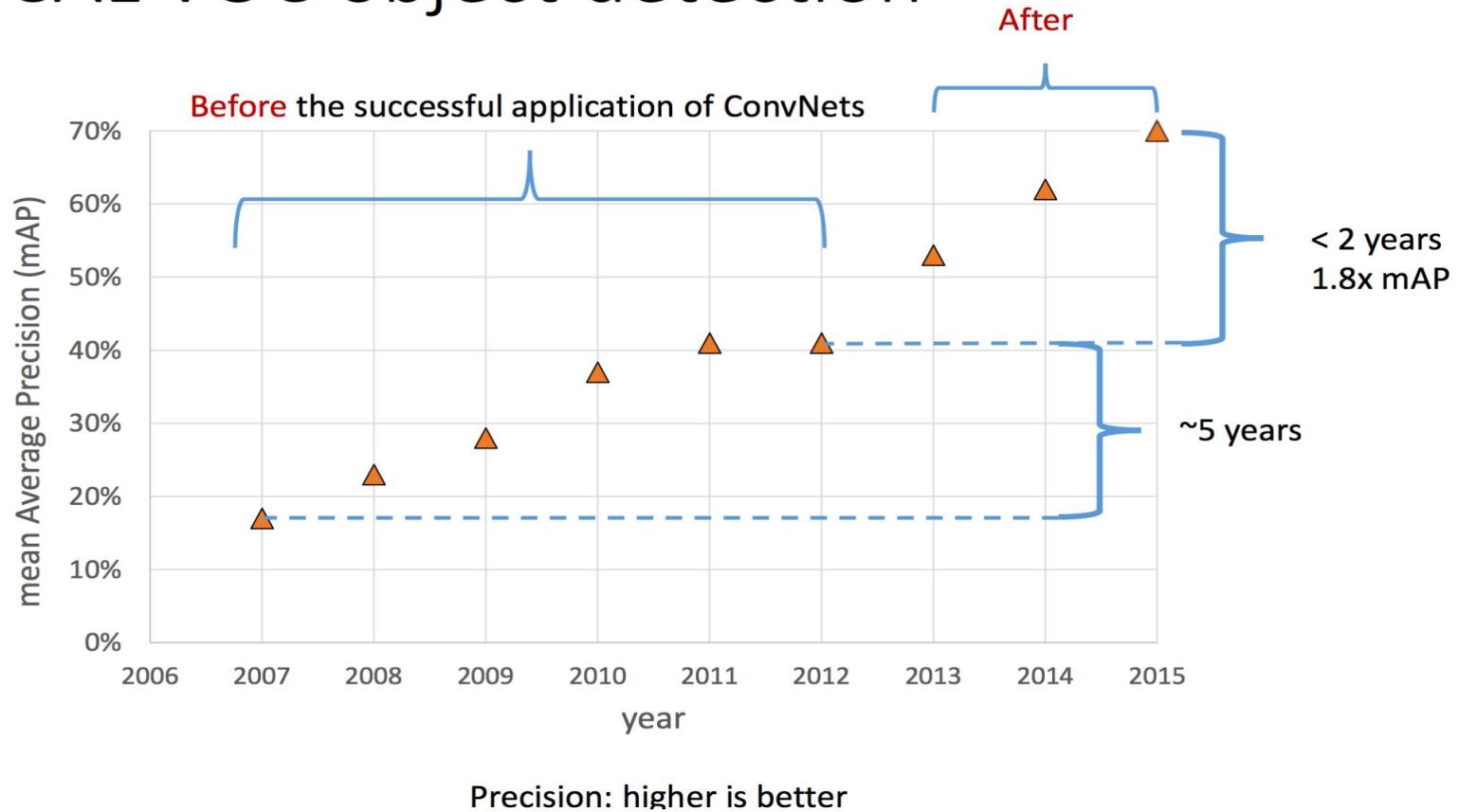


Object Detection



Huang et al., Speed/accuracy trade-offs for modern convolutional object detectors. CVPR 2017
figure/code at: https://github.com/tensorflow/models/tree/master/research/object_detection

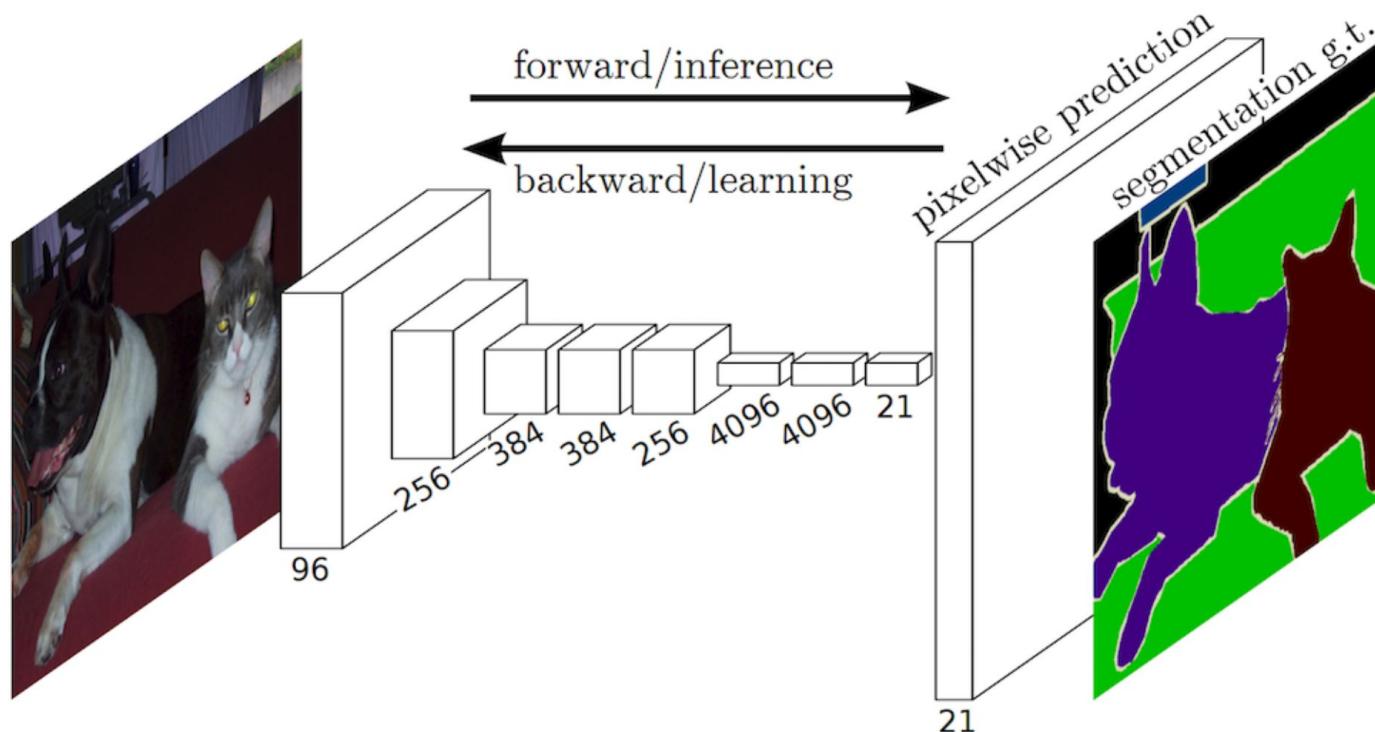
PASCAL VOC object detection



(Figur from Ross Girshick)

Semantic Segmentation

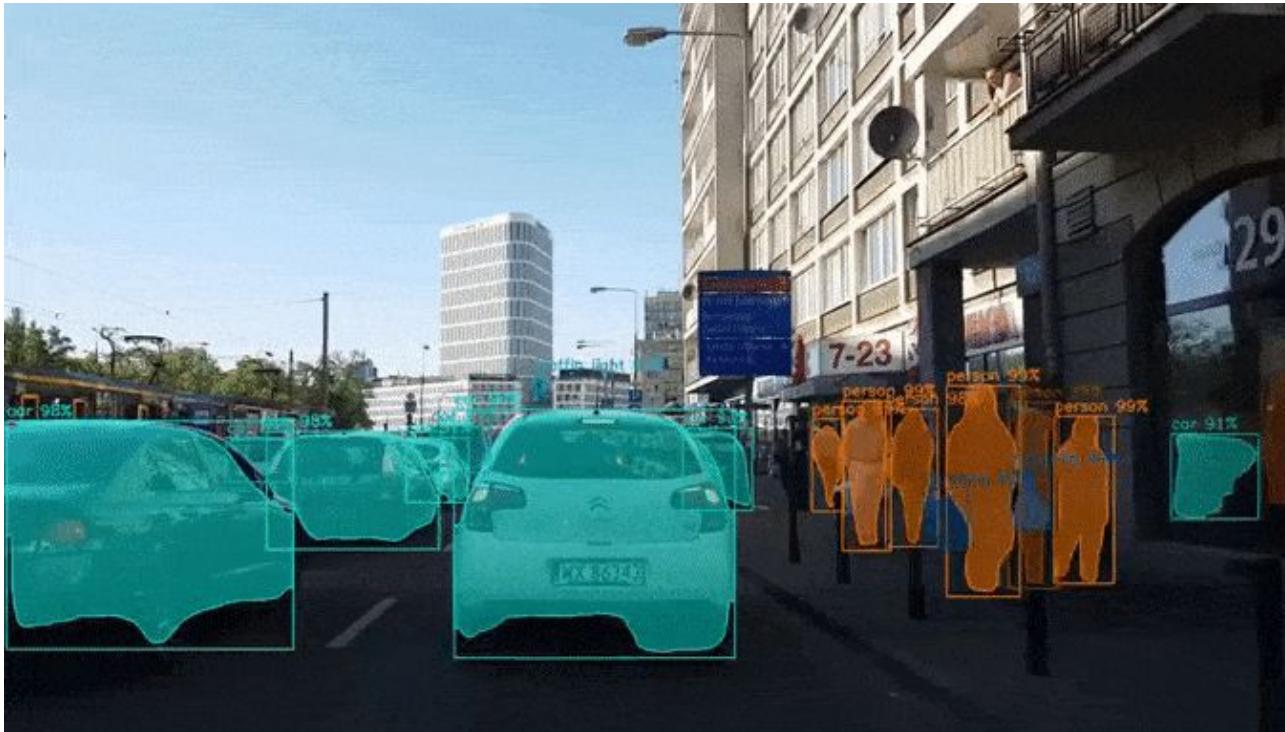
- Goal: Segment object regions and predict class labels for each region
- Can be formulated as pixel-wise classification



(Long et al, "Fully Convolutional Networks for Semantic Segmentation", CVPR, 2015.)

Instance Segmentation

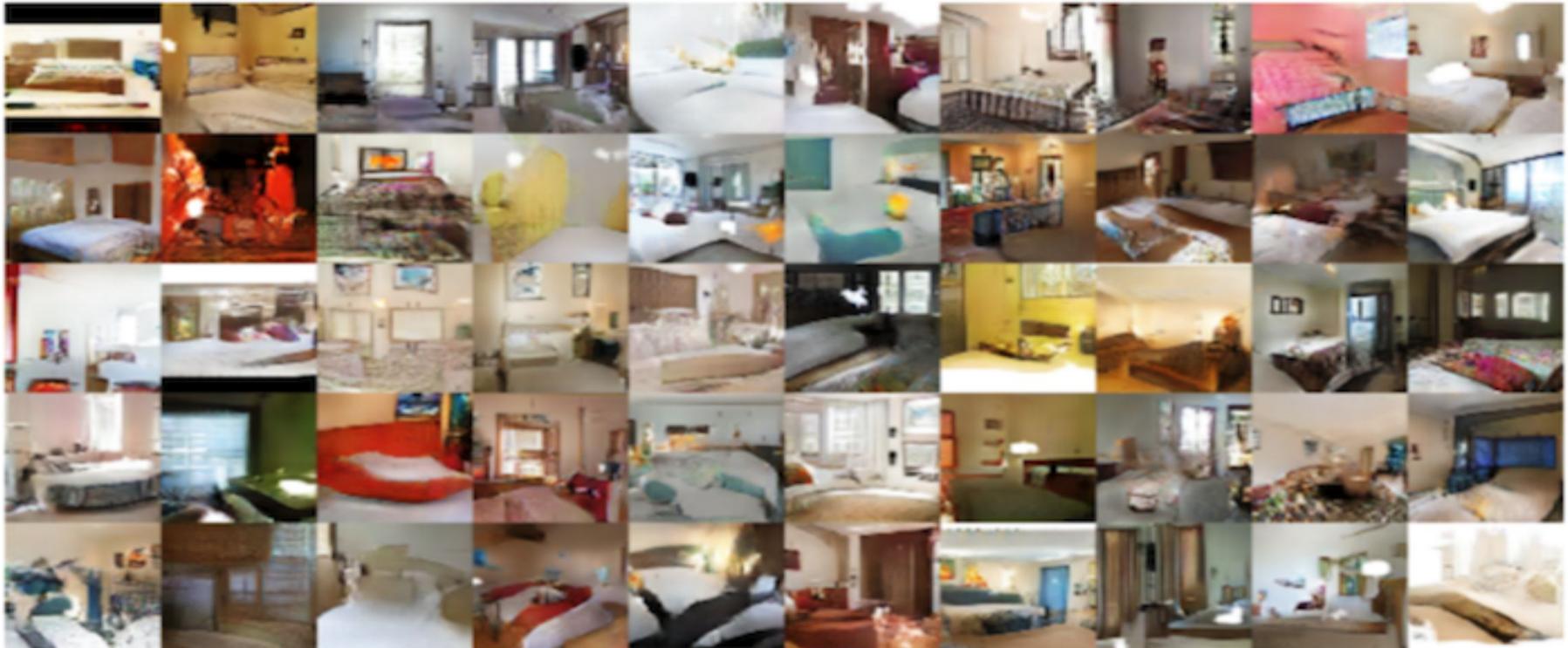
Goal: Perform segmentation (pixel-level masking) for each individual object



Mask R-CNN (He et al., 2017)

Image from: https://github.com/matterport/Mask_RCNN 64

Image Generation: Generative Adversarial Networks



(Radford et al, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", ICLR, 2016.)

Image Generation: Generative Adversarial Networks



Unsupervised Image-to-Image generation



Image Caption Generation

A group of young people playing a game of frisbee.



A herd of elephants walking across a dry grass field.



Two hockey players are fighting over the puck.



A close up of a cat laying on a couch.



A little girl in a pink hat is blowing bubbles.



A red motorcycle parked on the side of the road.



A refrigerator filled with lots of food and drinks.



A yellow school bus parked in a parking lot.



Describes without errors

Describes with minor errors

Somewhat related to the image

Unrelated to the image

(Vinyals et al, "Show and Tell: A Neural Image Caption Generator", CVPR, 2015.)

Text-to-Image Generation

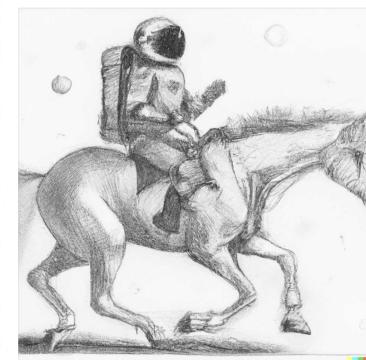
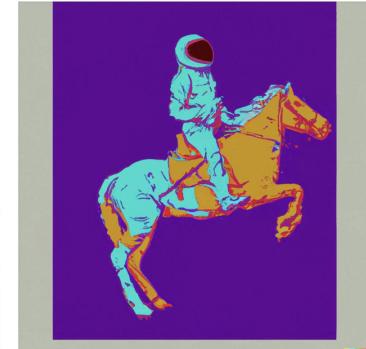
TEXT DESCRIPTION

An astronaut Teddy
bears A bowl of soup

riding a horse lounging in a
tropical resort in
space playing basketball
with cats in space

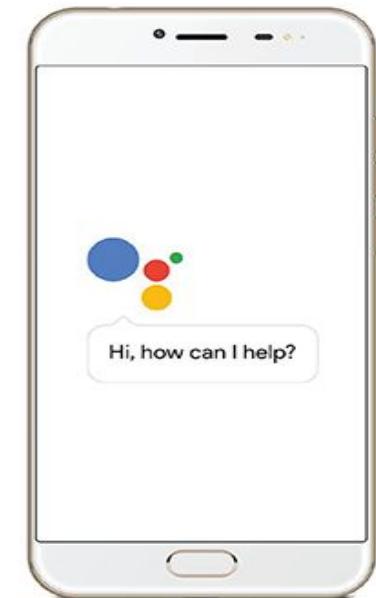
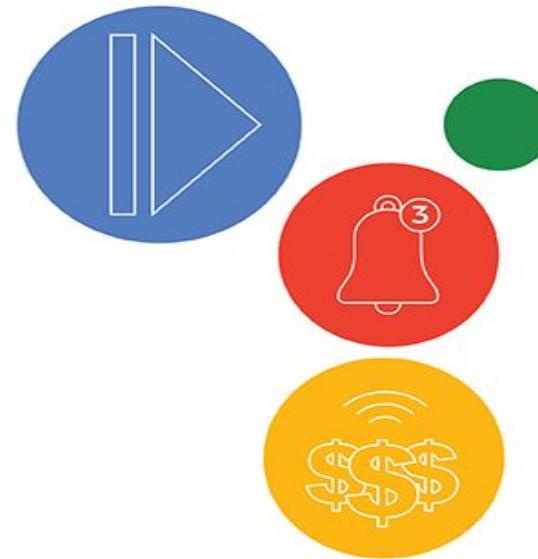
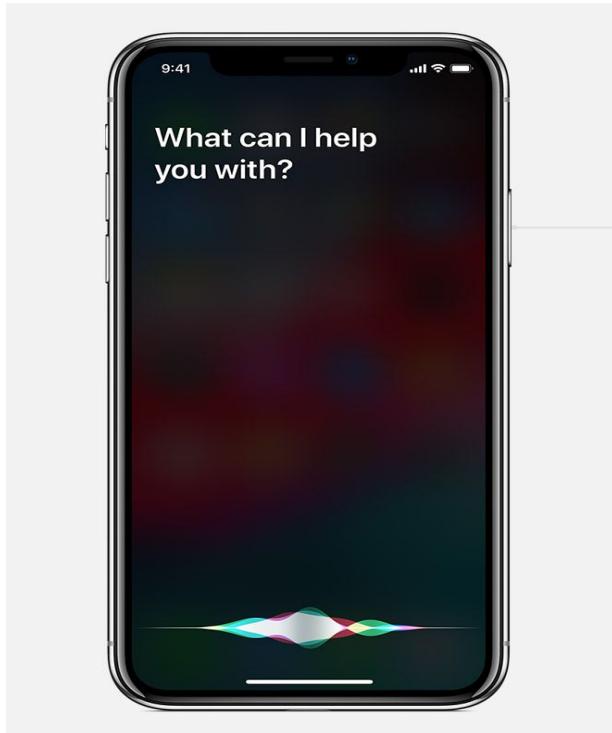
in a photorealistic style in
the style of Andy Warhol as
a pencil drawing

DALL·E 2



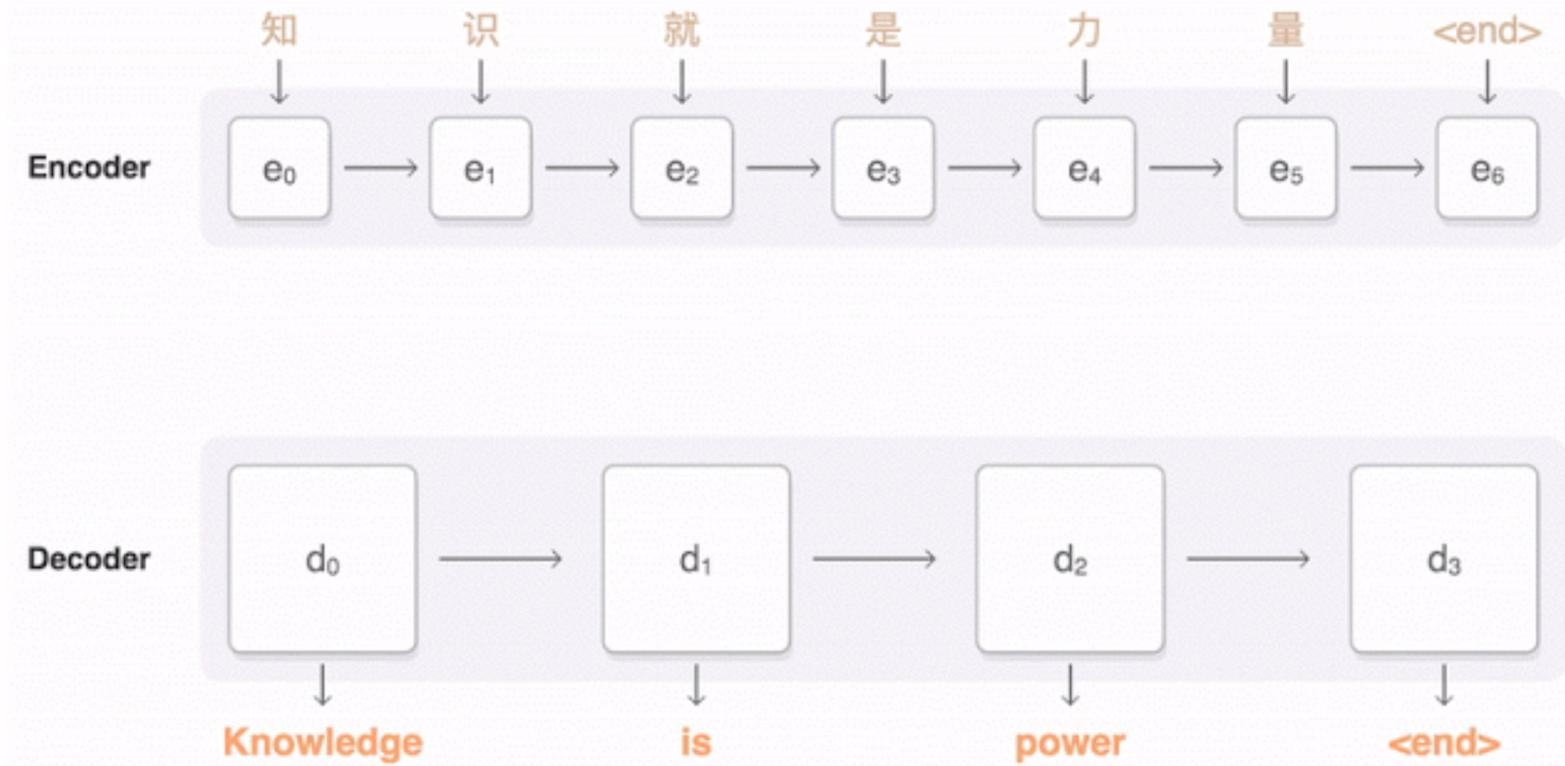
Speech recognition

Siri, Google home, and Google assistant achieves commercial-level performance



Machine Translation

Google Neural Machine Translation (in production)



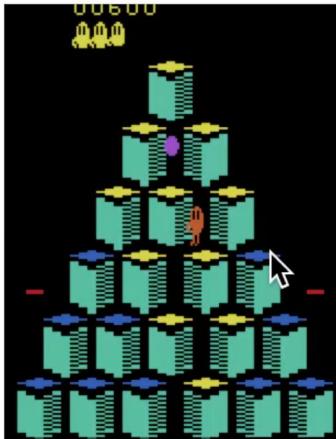
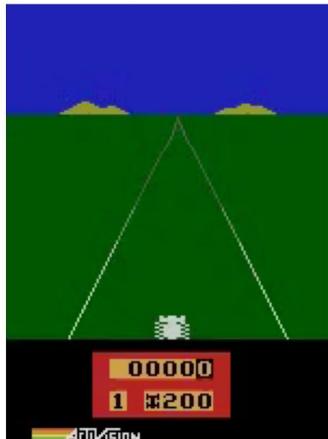
Large Language Models

SYSTEM PROMPT (HUMAN-WRITTEN)	<p><i>John F. Kennedy was just elected President of the United States after rising from the grave decades after his assassination. Due to miraculous developments in nanotechnology, Kennedy's brain was rebuilt from his remains and installed in the control center of a state-of-the art humanoid robot. Below is a transcript of his acceptance speech.</i></p>
MODEL COMPLETION (MACHINE-WRITTEN, 25 TRIES)	<p>"It is time once again. I believe this nation can do great things if the people make their voices heard. The men and women of America must once more summon our best elements, all our ingenuity, and find a way to turn such overwhelming tragedy into the opportunity for a greater good and the fulfillment of all our dreams. In the months and years to come, there will be many battles in which we will have to be strong and we must give all of our energy, not to repel invaders, but rather to resist aggression and to win the freedom and the equality for all of our people. The destiny of the human race hangs in the balance; we cannot afford for it to slip away. Now and in the years to come, the challenge before us is to work out how we achieve our ultimate destiny. If we fail to do so, we are doomed."</p> <p>"We, the descendants of the men and women of the Continental Army, who faced down a superior force on behalf of our independence and who fought against the evil powers of the world for the sake of freedom and human dignity, we, the heirs of men and women like Daniel Boone whom this country has produced, can accomplish our destiny as well as you have accomplished yours; and we pledge ourselves before the entire world, every second of every day."</p> <p>After the election, the President of the United States asked for a "more complete" body of information on Kennedy's brains' origin. In an e-mail exchange with me, Dr. Robert Wood, who, according to Wikipedia is the head of the University of Maryland Advanced Nanotechnology Research project, provided the following information:</p> <p>(1) The brain of JFK was harvested and reconstructed via tissue sampling. There was no way that the tissue could be transported by air. (2) A sample was collected from the area around his upper chest and sent to the University of Maryland for analysis. A human brain at that point would be about one and a half cubic centimeters. The data were then analyzed along with material that was obtained from the original brain to produce a reconstruction; in layman's terms, a "mesh" of brain tissue. There were no additional funds from the Department of Defense involved. The samples were sent back to the FBI lab for review and analysis. (3) There was never an attempt to clone any of America's greatest presidents. As far as we knew, the President was one of the most famous people on planet earth. If it had been possible, it would have.</p>

User	help me write a short note to introduce myself to my neighbor
ChatGPT	Hi there, My name is [Your Name] and I just moved in next door. I wanted to reach out and introduce myself. I'm looking forward to getting to know you and the neighborhood. If you ever need anything or just want to chat, please don't hesitate to stop by or give me a call. Best, [Your Name]
User	can you make it more formal?
ChatGPT	Dear Neighbor, I am writing to introduce myself. My name is [Your Name] and I have recently moved in next door. I wanted to take a moment to reach out and say hello. I am looking forward to getting to know you and the neighborhood. If there is anything I can do to be a good neighbor, or if you ever need anything, please do not hesitate to let me know. I am happy to help in any way that I can. Sincerely, [Your Name]
User	can you send it for me?
ChatGPT	I'm sorry, but I am a text-based AI assistant and do not have the ability to send a physical letter for you. I can only provide suggestions and advice on what to write. It is up to you to take that advice and send the letter yourself.

Examples from: OpenAI [GPT-2](#), [ChatGPT](#)

RL success stories: playing ATARI games



DQN Mnih et al, NIPS 2013 / Nature 2015;

MCTS Guo et al, NIPS 2014; **TRPO** Schulman, Levine, Moritz, Jordan, Abbeel, ICML 2015;
A3C Mnih et al, ICML 2016; **Dueling DQN** Wang et al ICML 2016; **Double DQN** van Hasselt et al, AAAI 2016; **Prioritized Experience Replay** Schaul et al, ICLR 2016; **Bootstrapped DQN** Osband et al, 2016; **Q-Ensembles** Chen et al, 2017; **Rainbow** Hessel et al, 2017; ...

AlphaGo

- Another breakthrough from Google DeepMind
- Combines Monte-Carlo Tree Search (MCTS) with deep neural networks



AlphaGo Silver et al, Nature 2015

AlphaGoZero Silver et al, Nature 2017

AlphaZero Silver et al, 2017

Tian et al, 2016; Maddison et al, 2014; Clark et al, 2015

OpenAI's 1v1 Dota [2017] and 5v5 [2018]

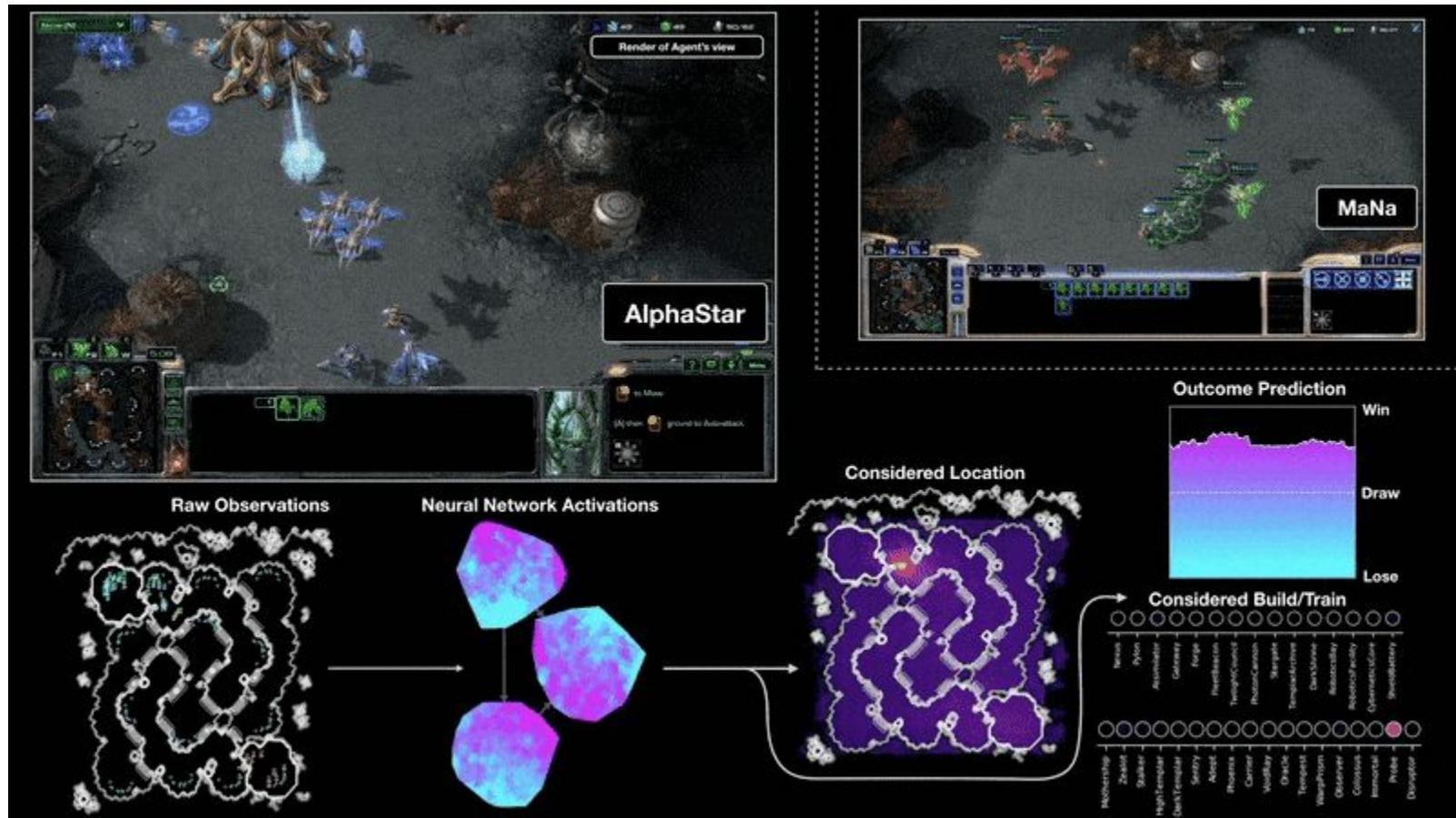
Super-human agent on a competitive game, enabled by

- Reinforcement learning
- Self-play
- Enough computation

Cooperation emerges



StarCraft II: DeepMind's AlphaStar



Vinyals et al., Grandmaster Level in StarCraft II Using Multi-agent Reinforcement Learning. 2019
Credit: <https://www.deepmind.com/blog/alphastar-mastering-the-real-time-strategy-game-starcraft-ii>

Robot learning



Levine et al., Learning Hand-Eye Coordination for Robotic Grasping. 2016
Video: https://www.youtube.com/watch?v=cXaic_k80uM slide credit: Pieter Abbeel

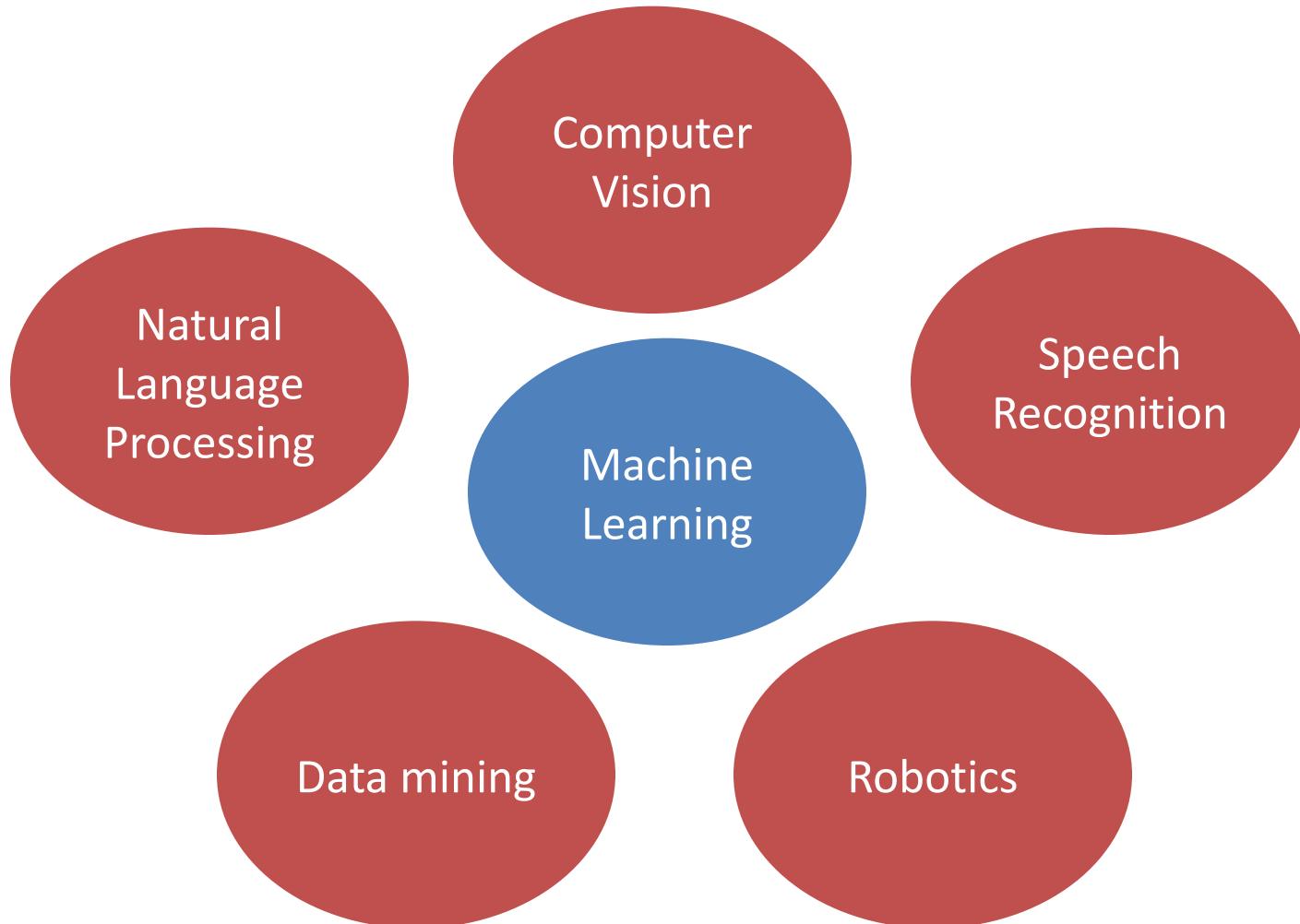
Self-driving cars



<https://youtu.be/O6DRfAC1JXA>

See also: Chris Urmson: How a driverless car sees the road <https://youtu.be/tiwVMrTLUWg>

Machine Learning and other fields



Next class

- Supervised learning
 - Linear regression

Reminder

- Check syllabus at Canvas
- For all questions, please use Piazza (linked to Canvas)

Questions?

Any feedback (about lecture, slide, homework, project, etc.)?

(via anonymous google form: <https://forms.gle/99jeftYTaozJvCEF8>)



Change Log of lecture slides:

https://docs.google.com/document/d/e/2PACX-1vRKx40eOJKACqrKWraio0AmIFS1_xBMINuWcc-jzpfo-ySj_gBuqTVdfHy8v4HDmqDJ3b3TvAW1FVuH/pub