Warm up as you log in

Welcome!

1) Rename yourself in Zoom to append your house name
   - e.g. Pat Virtue – House 0
   - If you don't yet know your house, no change needed ☺
   - If you aren't on Slack, see Piazza "Slack Link for the Houses"
2) Take student survey
   - See Piazza "Student Survey"
3) Explore Zoom
   - Find hand-raise icon in Participants
   - Find Chat

Introduction to Machine Learning

Instructor: Pat Virtue

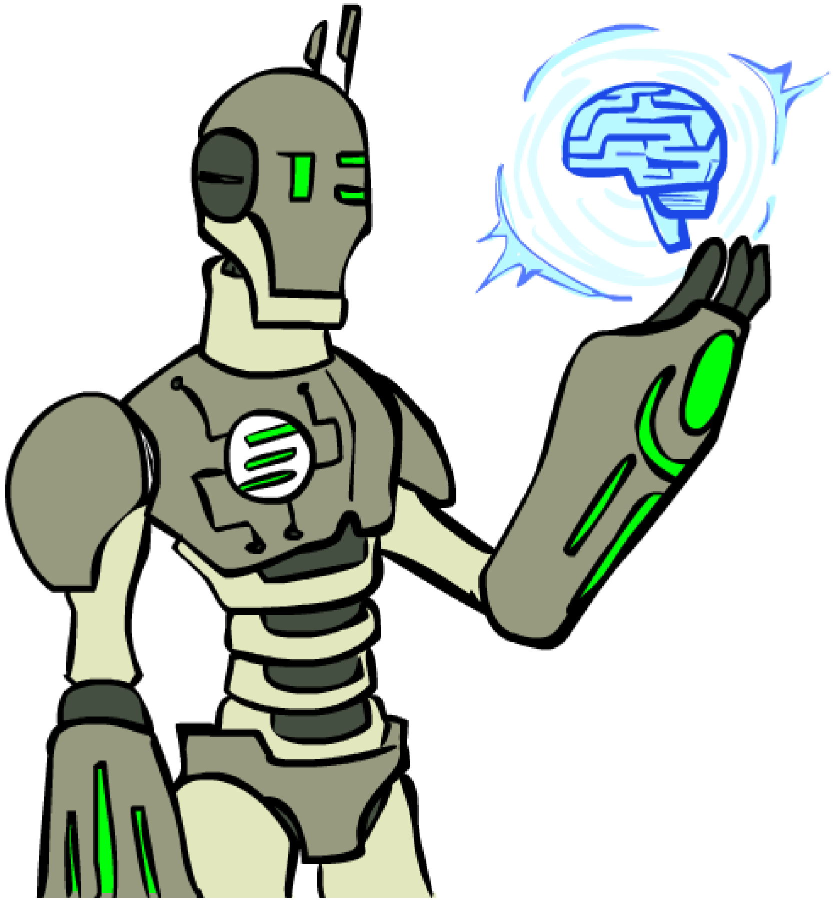
Today

Course Info

AI /ML Definitions and History

ML Problem Formulation

More Course Info



Images: ai.berkeley.edu

Breakout Rooms

Always

1. Video on
2. Unmute
3. Introduce yourself

Now

1. Start a conversation
   a) What is your major / program of study
   b) Are you new to CMU?

Course Information

Website: https://www.cs.cmu.edu/~10601

Canvas: canvas.cmu.edu

Gradescope: gradescope.com

Communication:

piazza.com

E-mail (if piazza doesn't work):

EAs-10601-2020@mailman.srv.cs.cmu.edu

pvirtue@andrew.cmu.edu

Course Information

Lectures, Recitations, Office hours:

- Lectures are recorded (but not recitation or OH)
  - Shared with our course and ML course staff only
  - Breakout rooms are not recorded
- If you have a question:
  - Use the Hand-raise icon in the participant window
- If we're asking you a question
  - Use the Hand-raise icon in the participant window
  - Or, just unmute and answer
- If you're talking:
  - Turn your video on ☺
  - Especially in cases when we are not recording

Course Information

Lectures and Recitations

- You can attend any lecture section you want
- You can attend any recitation section you want
- Please register in SIO for the lecture and recitation you plan to attend
- Midterm exams will be in lecture section

Participation Points

- Primarily earned by answering Piazza Polls in live lecture
- There are a few other ways to earn points too, stay tuned to Piazza for details

Houses

Slack

- Communication within your House
- See Piazza for invite link
- Don't worry if you haven't been placed in your House Slack channel yet

Houses

- Smaller group of students
  - House spirit, Share concerns, Form study groups, Work on side projects
- TA for each house

The Sorting $\hat{y}$

Course Team

Education Associates

Joshmin Ray
joshminr

Fatima Kizilkaya
fjeffrey

Brynn Edmunds
bedmunds

Course Team

Teaching Assistants

Varsha
vkuppurr

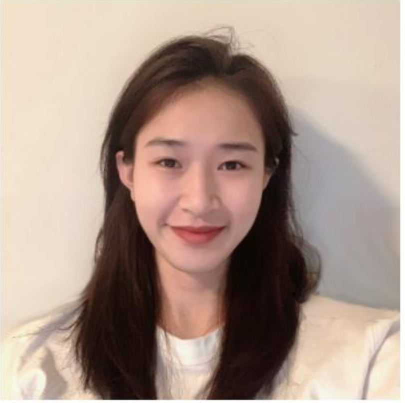
Hanyue
hanyuech

Andrew
andrewh1

Zhaomin
zhaominz

Everett
eknag

Alex
alexs1



Nan
nany

Adrian
akager

Course Team

Teaching Assistants

Scott
sicongli

Laura
yurongl

Hongyi
hongyiz2

Daniel
seungwom

Young
youngwo1

Zhengyang
zhengyax

Ani
achowdh1

Eric
esliang

Course Team

Instructor

Pat
Virtue
pvirtue

Students!!

AI in the News

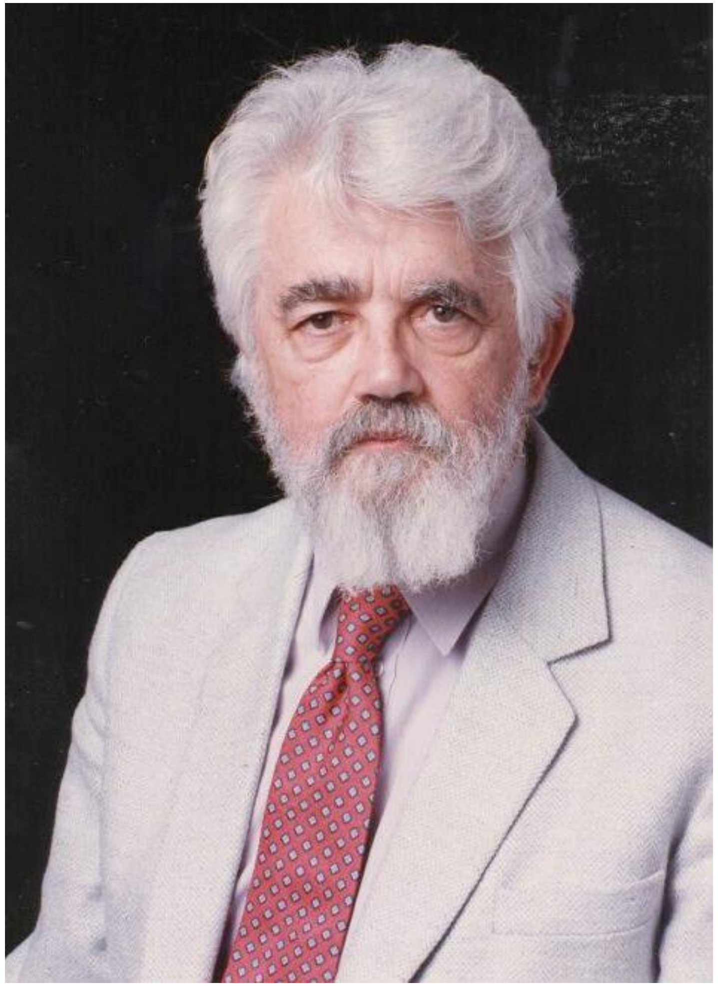
AI Definition by John McCarthy

What is artificial intelligence

- It is the science and engineering of making intelligent machines, especially intelligent computer programs

What is intelligence

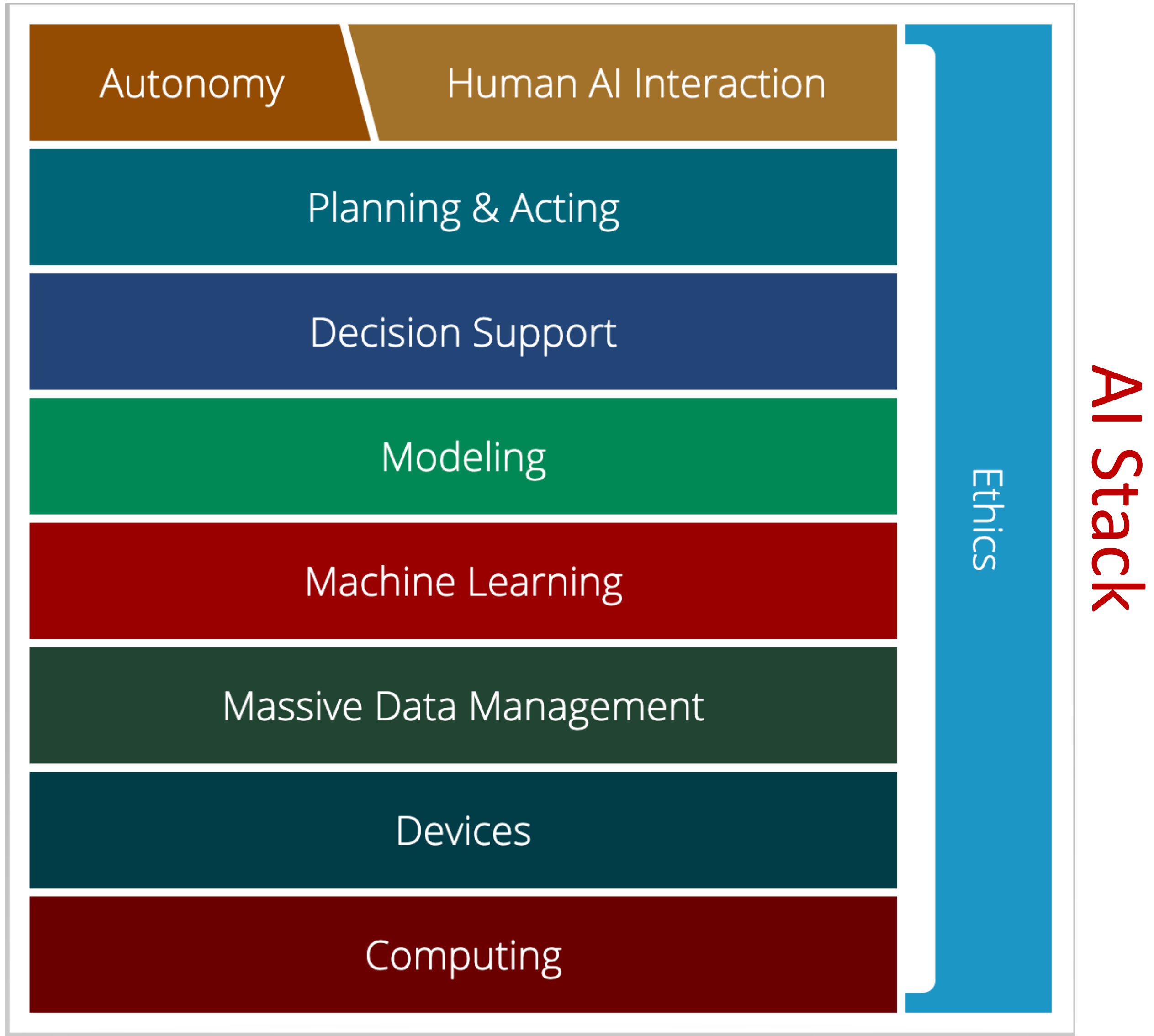
- Intelligence is the computational part of the ability to achieve goals in the world



http://www-formal.stanford.edu/jmc/whatisai/whatisai.html

AI Stack for CMU AI

"AI must understand the human needs and it must make smart design decisions based on that understanding"

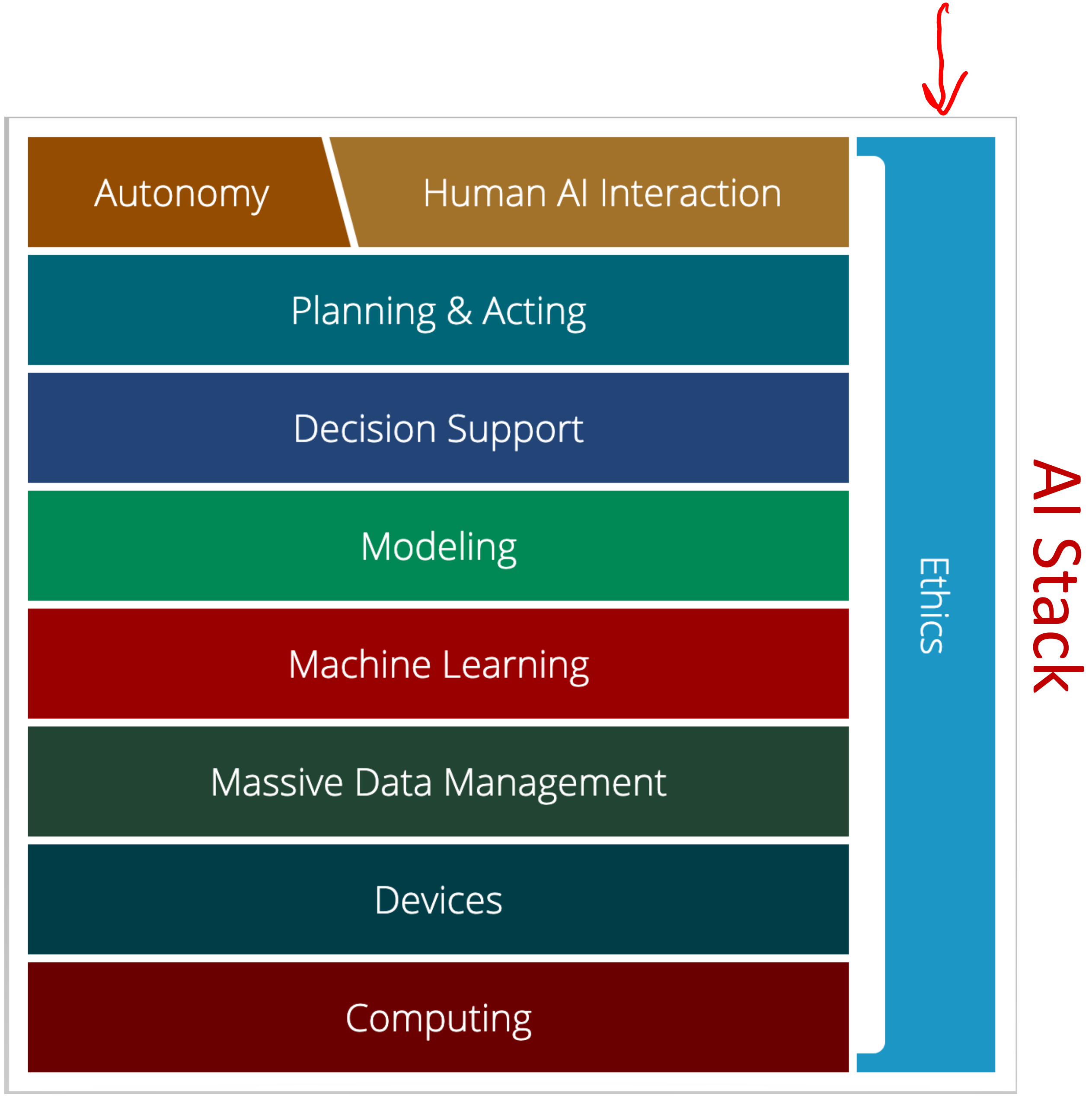


https://ai.cs.cmu.edu/about

AI Stack for CMU AI

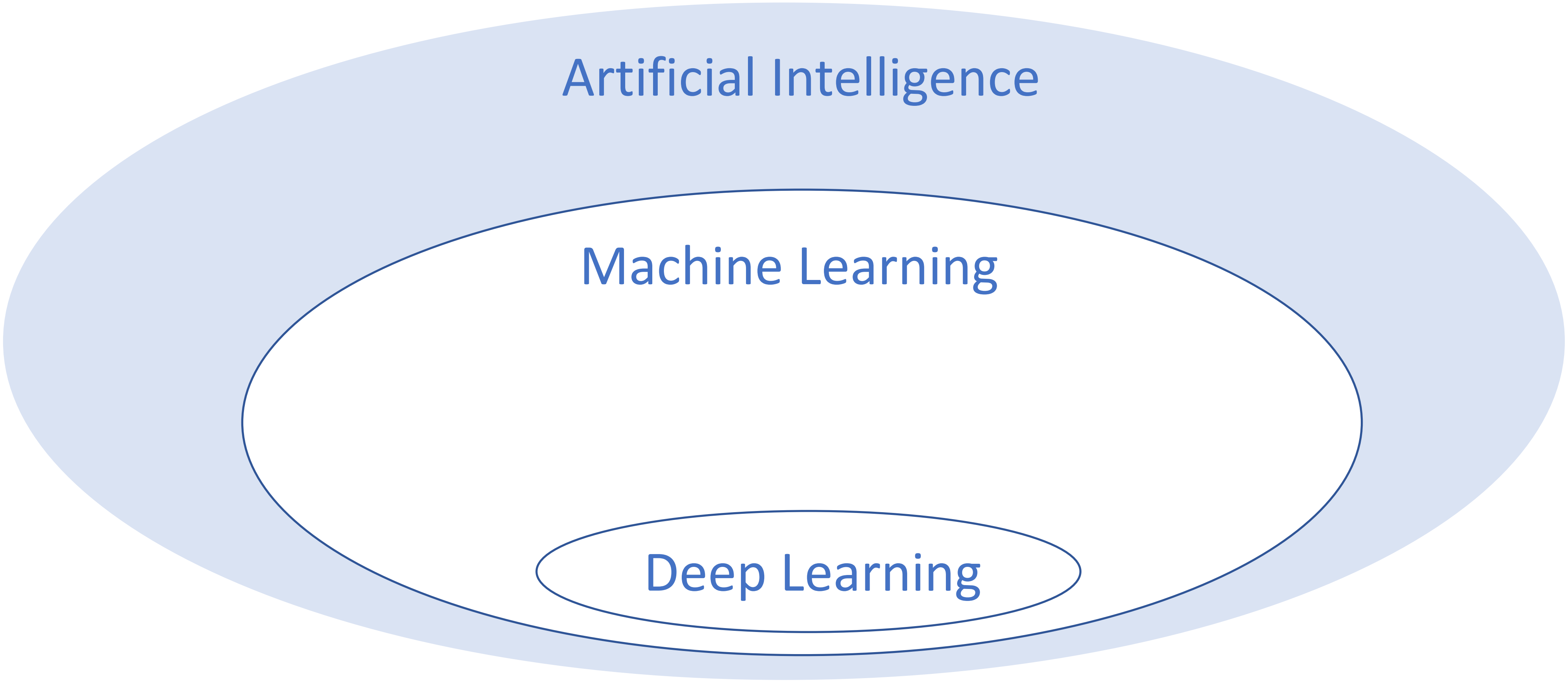
"Machine learning focuses on creating programs that learn from experience."

"It advances computing through exposure to new scenarios, testing and adaptation, while using pattern- and trend-detection to help the computer make better decisions in similar, subsequent situations."

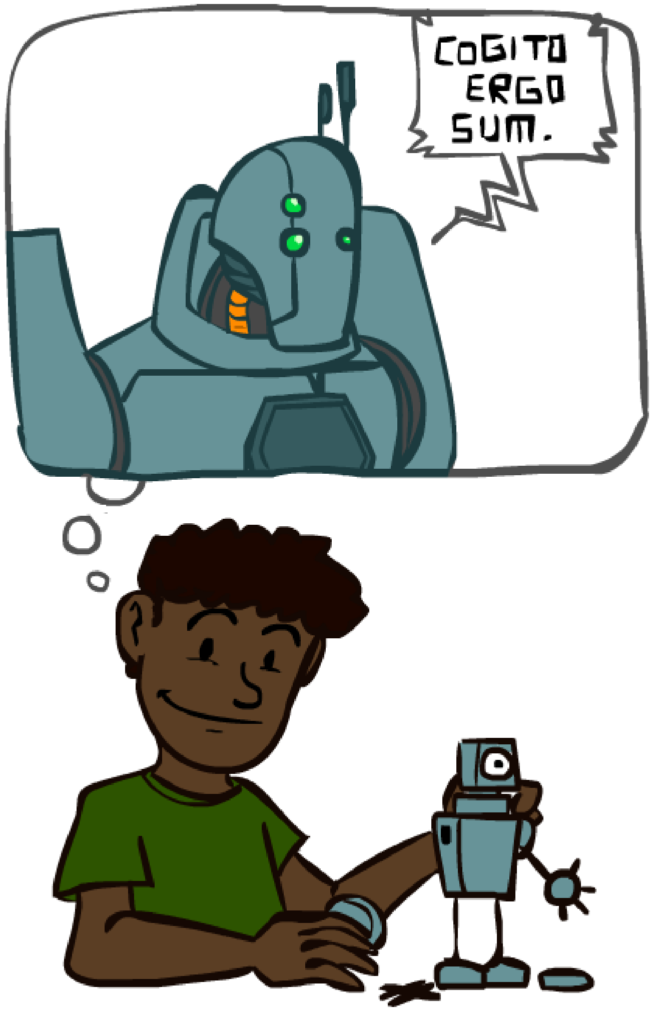


https://ai.cs.cmu.edu/about

Artificial Intelligence vs Machine Learning?



A Brief History of AI



Images: ai.berkeley.edu

A Brief History of AI

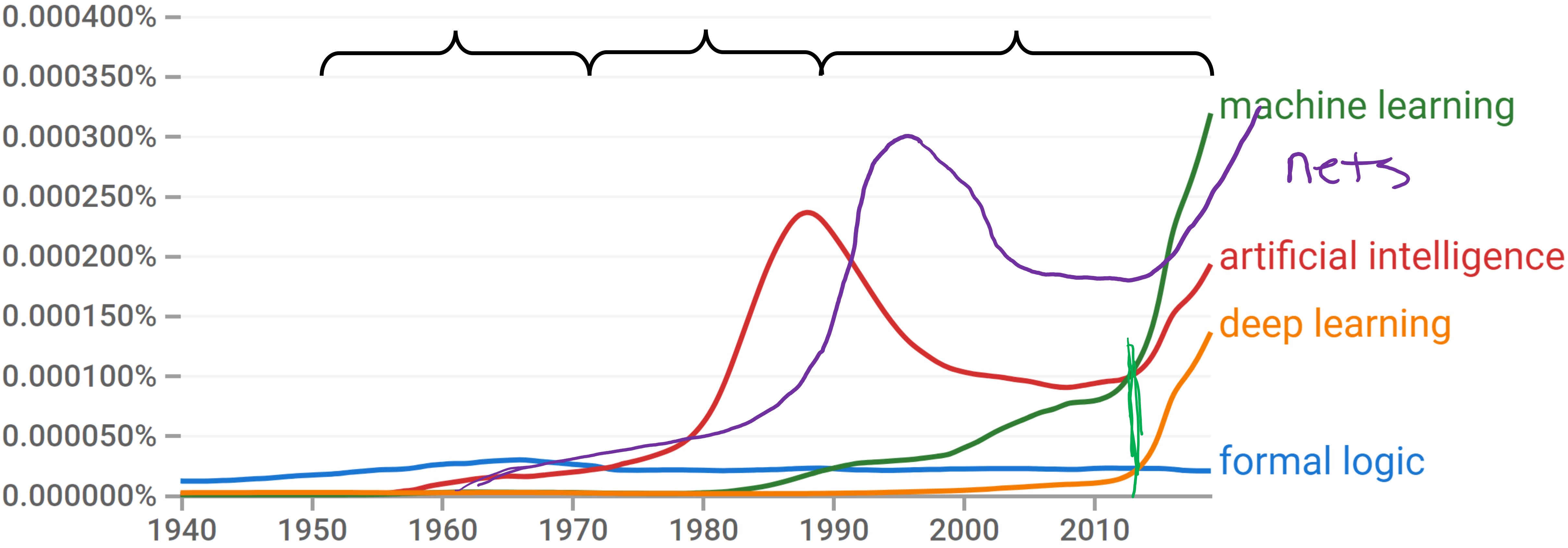
n-grams

AI Excitement! 1950-1970

Knowledge Based Systems 1970-1990

Statistical Approaches 1990-

Deep Learning Era 2012-



https://books.google.com/ngrams

A Brief History of AI

1940-1950: Early days

- 1943: McCulloch & Pitts: Boolean circuit model of brain
- 1950: Turing's "Computing Machinery and Intelligence"

1950—70: Excitement: Look, Ma, no hands!

- 1950s: Early AI programs, including Samuel's checkers program, Newell & Simon's Logic Theorist, Gelernter's Geometry Engine
- 1956: Dartmouth meeting: "Artificial Intelligence" adopted

1970—90: Knowledge-based approaches

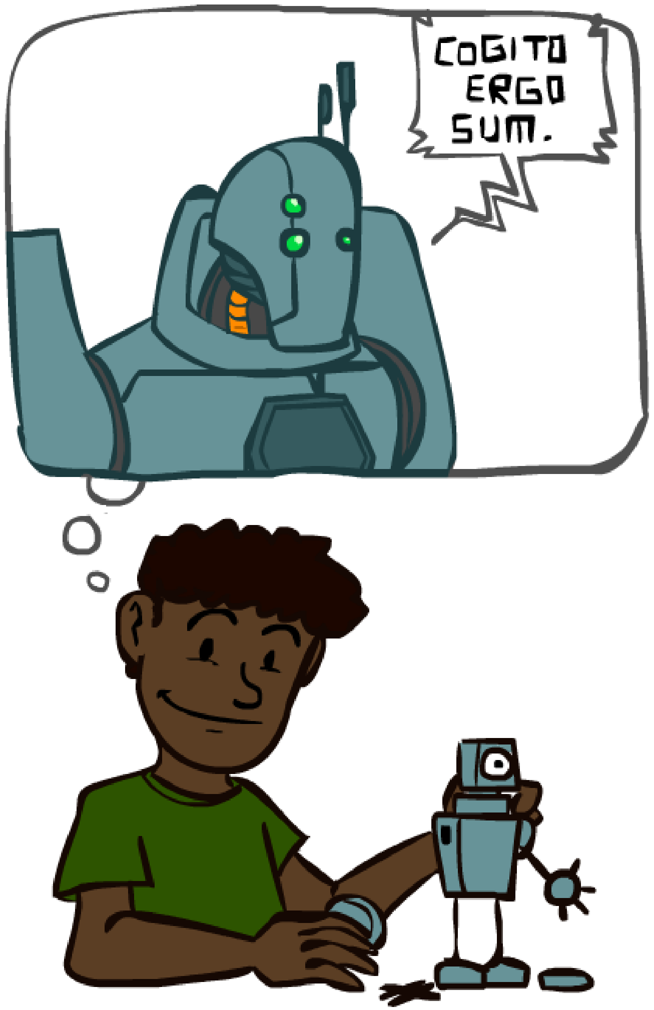
- 1969—79: Early development of knowledge-based systems
- 1980—88: Expert systems industry booms
- 1988—93: Expert systems industry busts: "AI Winter"

1990—: Statistical approaches

- Resurgence of probability, focus on uncertainty
- General increase in technical depth
- Agents and learning systems… "AI Spring"?

2012—: Deep learning

- 2012: ImageNet & AlexNet



Images: ai.berkeley.edu

Breakout Rooms

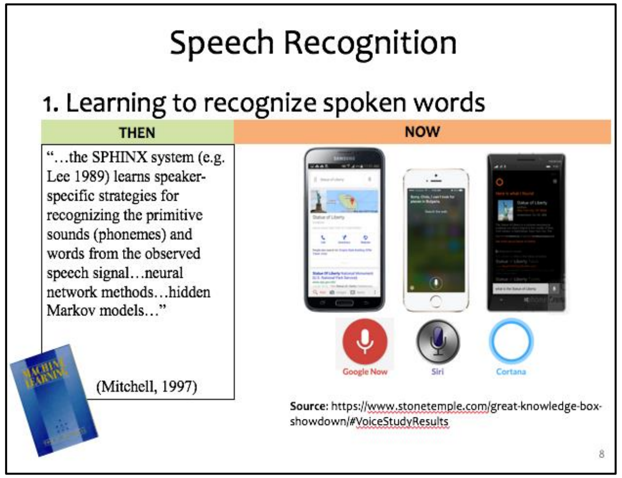
Always

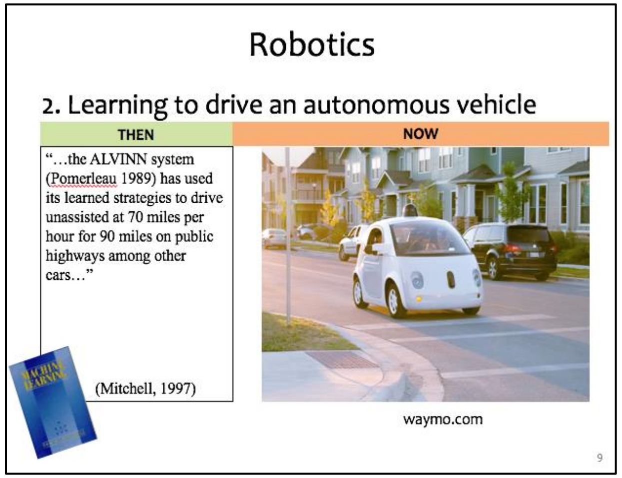
1. Video on
2. Unmute
3. Introduce yourself

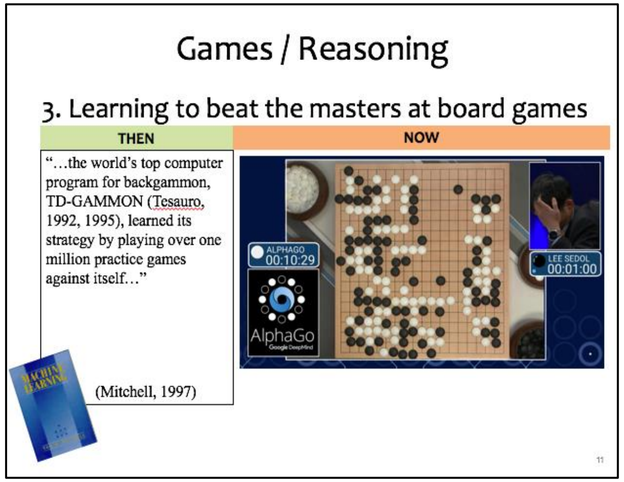
Now

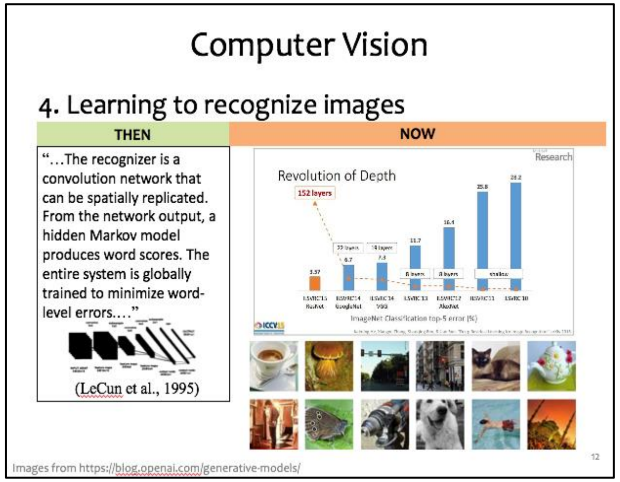
1. Are we headed for another AI winter?

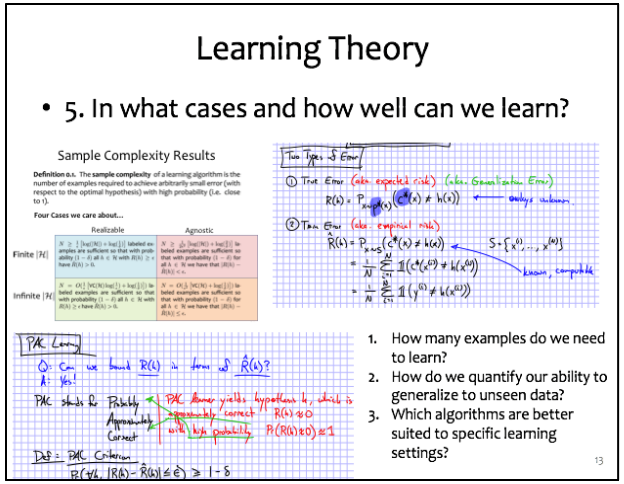
ML Applications?





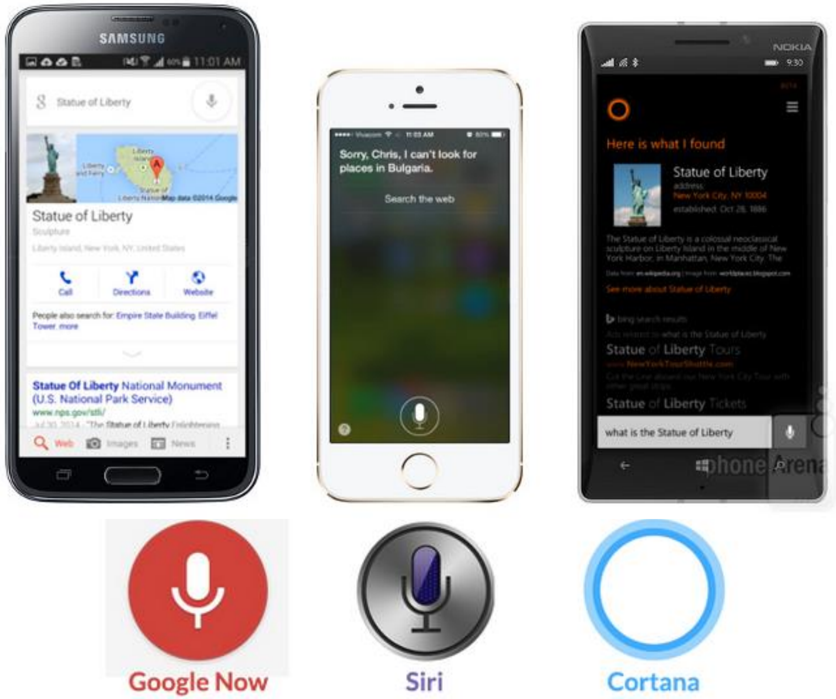


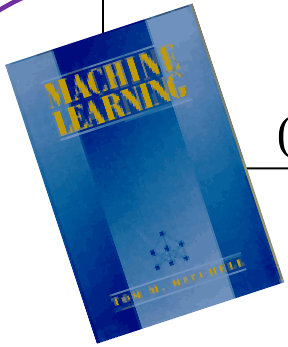




Speech Recognition

1. Learning to recognize spoken words

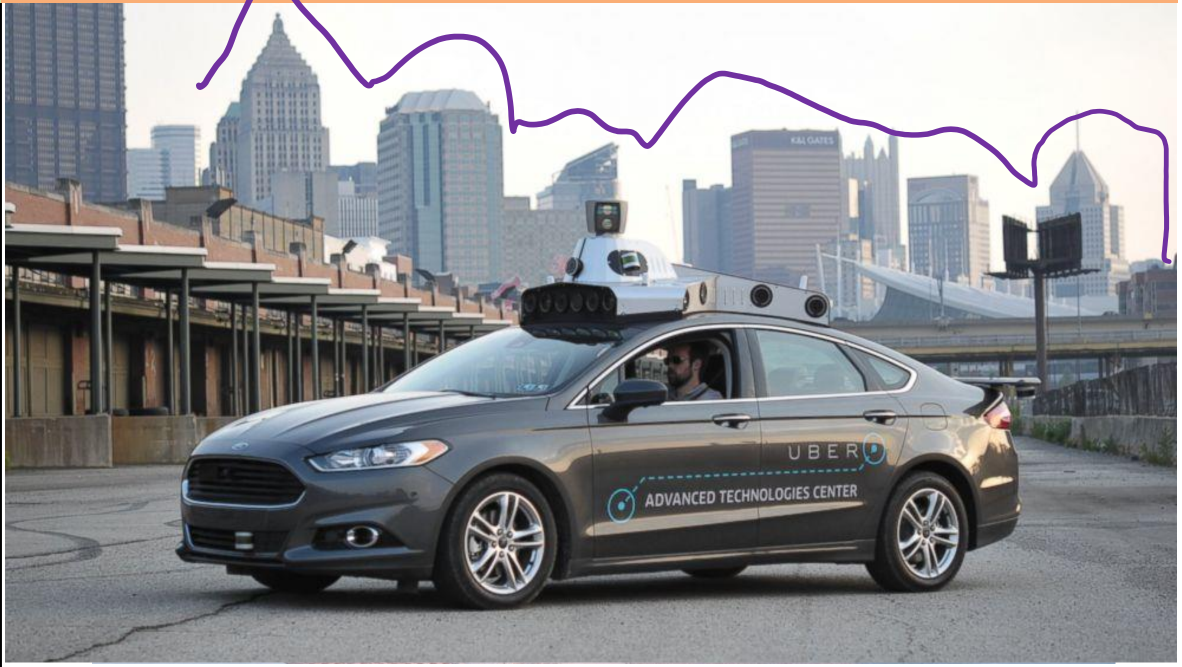
THEN	NOW
"…the SPHINX system (e.g. Lee 1989) learns speaker-specific strategies for recognizing the primitive sounds (phonemes) and words from the observed speech signal…neural network methods…hidden Markov models…" (Mitchell, 1997)	

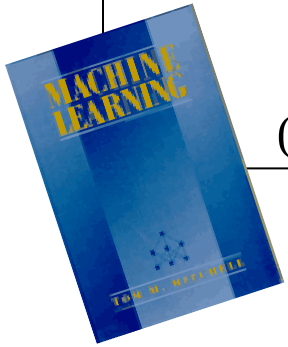


Source: https://www.stonetemple.com/great-knowledge-box-showdown/#VoiceStudyResults

Robotics

2. Learning to drive an autonomous vehicle

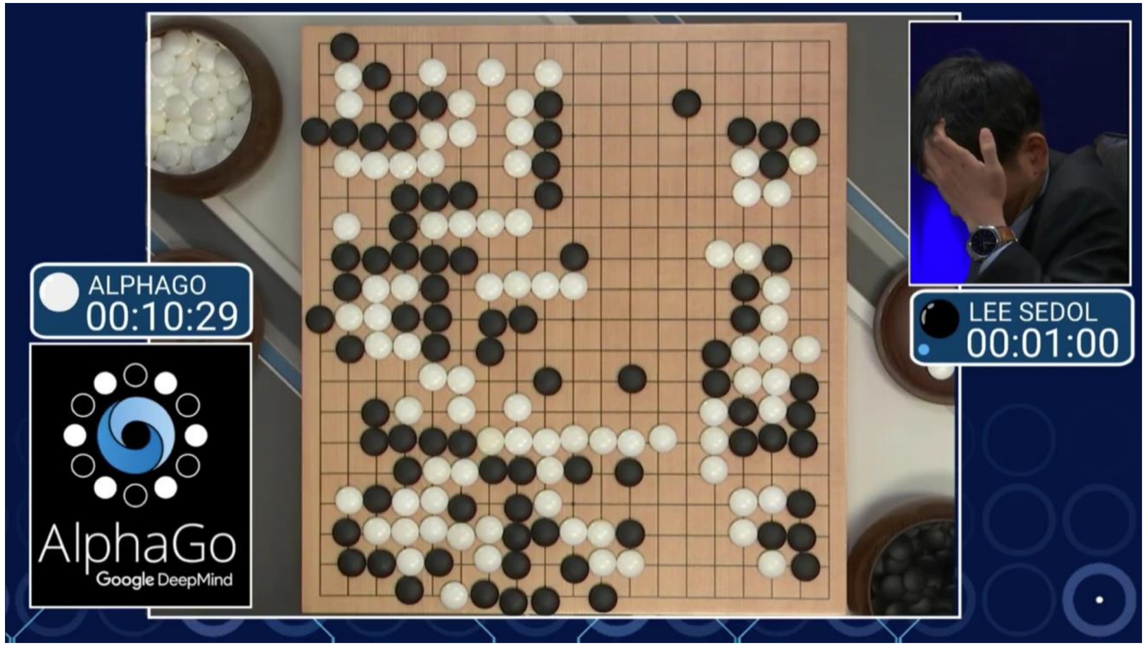
THEN	NOW
"…the ALVINN system (Pomerleau 1989) has used its learned strategies to drive unassisted at 70 miles per hour for 90 miles on public highways among other cars…" (Mitchell, 1997)	

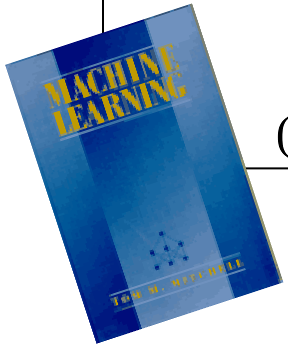


https://www.geek.com/wp-content/uploads/2016/03/uber.jpg

Games / Reasoning

3. Learning to beat the masters at board games

THEN	NOW
"…the world's top computer program for backgammon, TD-GAMMON (Tesauro, 1992, 1995), learned its strategy by playing over one million practice games against itself…" (Mitchell, 1997)	



Computer Vision

4. Learning to recognize images

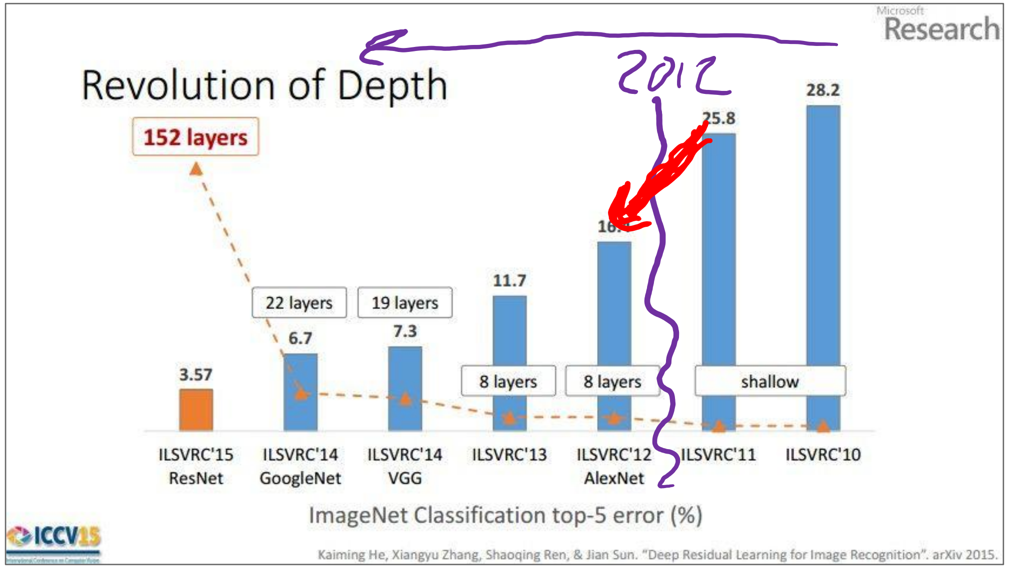
THEN	NOW

"…The recognizer is a convolution network that can be spatially replicated. From the network output, a hidden Markov model produces word scores. The entire system is globally trained to minimize word-level errors.…"

Figure 2: Convolutional neural network character recognizer. This architecture is robust to local translations and distortions, with subsampling, shared weights, and local receptive fields.

number of subsampling layers and the sizes of the kernels are chosen, the sizes of all the layers, including the input, are determined unambiguously. The only architectural parameters that remain to be selected are the number of feature maps in each layer, and the information as to what feature map is connected to what other feature map. In our case, the subsampling rates were chosen as small as possible (2 × 2), and the kernels as small as possible in the first layer (3 × 3) to limit the total number of connections. Kernel sizes in the upper layers are chosen to be as small as

(LeCun et al., 1995)



Images from https://blog.openai.com/generative-models/

Learning Theory

- 5. In what cases and how well can we learn?

Sample%Complexity%Results

Definition 0.1. The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four$Cases$we$care$about...

	Realizable	Agnostic
Finite $\|\mathcal{H}\|$	$N \geq \frac{1}{\epsilon}[\log(\|\mathcal{H}\|)+\log(\frac{1}{\delta})]$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $R(h) \geq \epsilon$ have $\hat{R}(h) > 0$.	$N \geq \frac{1}{2\epsilon^2}[\log(\|\mathcal{H}\|)+\log(\frac{2}{\delta})]$ labeled examples are sufficient so that with probability $(1-\delta)$ for all $h \in \mathcal{H}$ we have that $\|R(h)-\hat{R}(h)\| < \epsilon$.
Infinite $\|\mathcal{H}\|$	$N = O(\frac{1}{\epsilon}[\mathrm{VC}(\mathcal{H})\log(\frac{1}{\epsilon})+\log(\frac{1}{\delta})])$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $R(h) \geq \epsilon$ have $\hat{R}(h) > 0$.	$N = O(\frac{1}{\epsilon^2}[\mathrm{VC}(\mathcal{H})+\log(\frac{1}{\delta})])$ labeled examples are sufficient so that with probability $(1-\delta)$ for all $h \in \mathcal{H}$ we have that $\|R(h)-\hat{R}(h)\| \leq \epsilon$.

Two Types of Error

① True Error (aka. expected risk) (aka. Generalization Error)

$$R(h) = P_{x \sim p^*(x)}(c^*(x) \neq h(x))$$ ← always unknown.

② Train Error (aka. empirical risk)

$$\hat{R}(h) = P_{x \sim S}(c^*(x) \neq h(x))$$ $S = \{x^{(1)}, \dots, x^{(N)}\}$

$$= \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(c^*(x^{(i)}) \neq h(x^{(i)}))$$ known, computable

$$= \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(y^{(i)} \neq h(x^{(i)}))$$

PAC Learning

Q: Can we bound $R(h)$ in terms of $\hat{R}(h)$?
A: Yes!

PAC stands for Probably Approximately Correct

PAC learner yields hypothesis h, which is approximately correct $R(h) \approx 0$ with high probability $\Pr(R(h) \approx 0) \approx 1$

Def: PAC Criterion

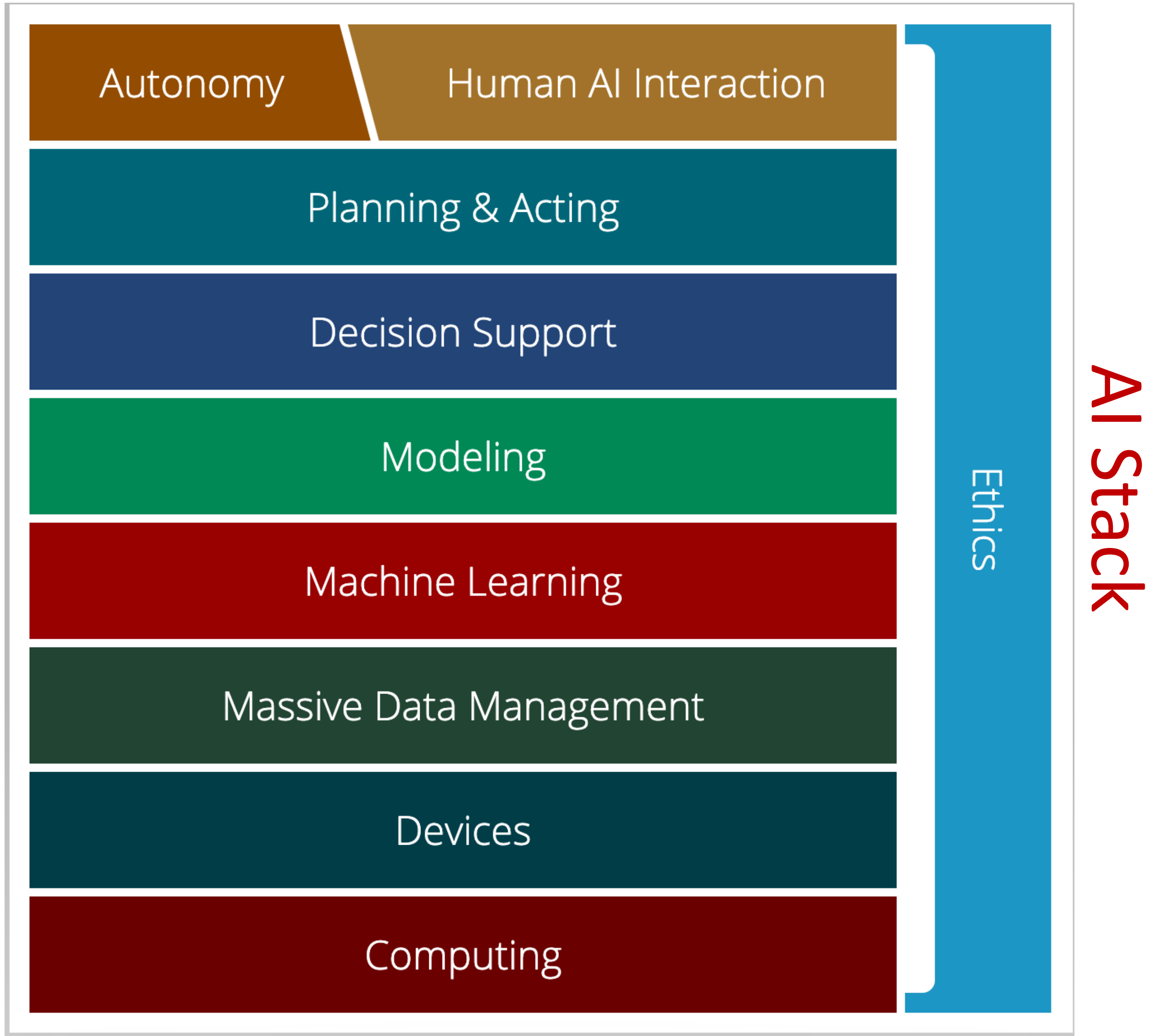
$$\Pr(\forall h, |R(h) - \hat{R}(h)| \leq \epsilon) \geq 1 - \delta$$

1. How many examples do we need to learn?
2. How do we quantify our ability to generalize to unseen data?
3. Which algorithms are better suited to specific learning settings?

AI Stack for CMU AI

“Machine learning focuses on creating programs that learn from experience.”

“It advances computing through exposure to new scenarios, testing and adaptation, while using pattern- and trend-detection to help the computer make better decisions in similar, subsequent situations.”



https://ai.cs.cmu.edu/about

DEFINING LEARNING PROBLEMS

Well-Posed Learning Problems

Three components *<T,P,E>***:**

1. Task, *T*
2. Performance measure, *P*
3. Experience, *E*

Definition of learning:

A computer program **learns** if its performance at tasks in *T*, as measured by *P*, improves with experience *E*.

Definition from (Mitchell, 1997)

Example Learning Problems

Learning to **beat the masters at chess**

1. Task, *T*:

2. Performance measure, *P*:

3. Experience, *E*:

Example Learning Problems

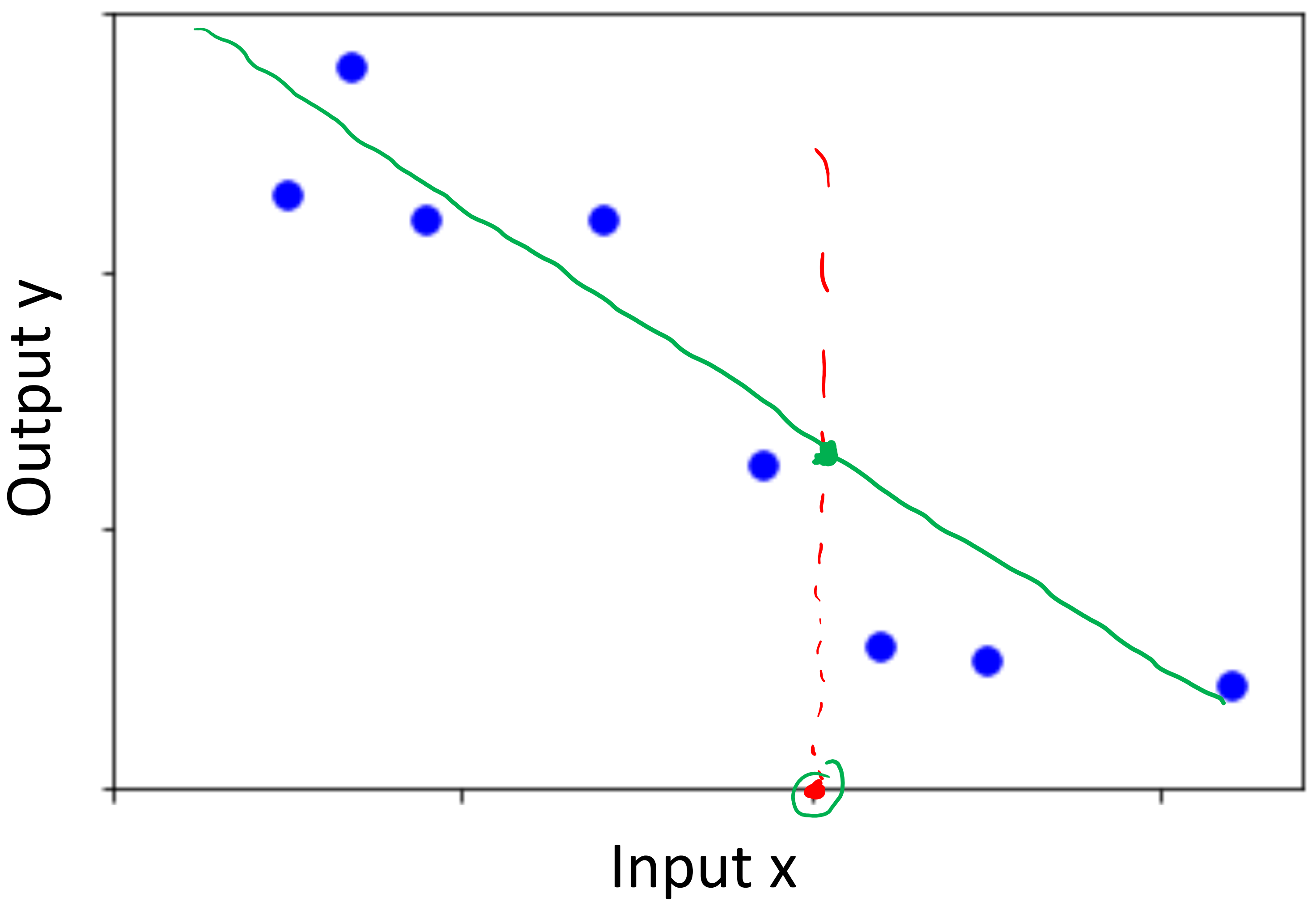
Learning to **respond to voice commands (Siri)**

1. Task, *T*:

2. Performance measure, *P*:

3. Experience, *E*:

Data: Input, Output, and Assumptions



Input:

- Discrete/Continuous?
- Single/Multiple values?

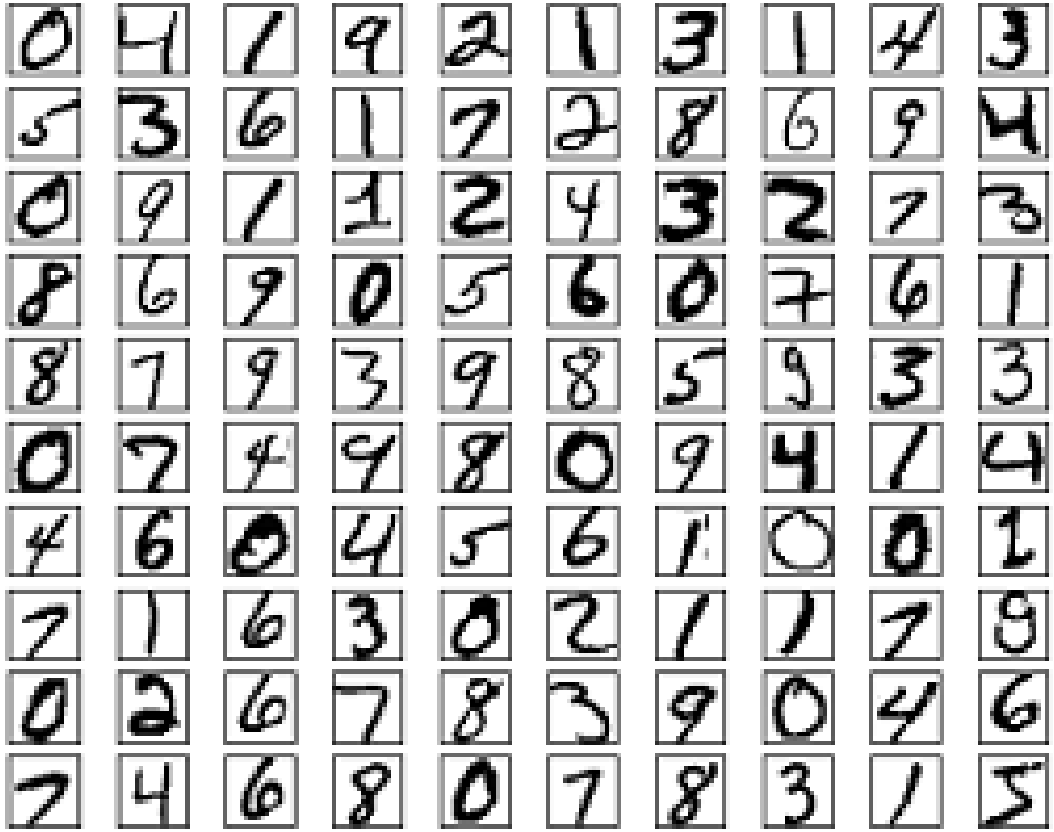
Output:

- Discrete/Continuous?
- Single/Multiple values?

Assumptions?

Data: Input, Output, and Assumptions

MNIST: Handwritten digits



Input:

- Discrete/Continuous?
- Single/Multiple values?

Output:

- Discrete/Continuous?
- Single/Multiple values?

Assumptions?

More robust examples: http://yann.lecun.com/exdb/lenet/index.html

Assumptions

Face dataset

Breakout Rooms

Search internet to find:

1. ML Classification Dataset
   - Discrete/unordered output
2. ML Regression Dataset
   - Continuous output

Are the input values discrete/continuous?

Are the input values a single value or multiple values?

What assumptions might we make with these datasets?

Course Information

Website: https://www.cs.cmu.edu/~10601

- Course Levels and Course Scope
- Prerequisites
- Teamwork
- Mental health