CS 4700: Foundations of Artificial Intelligence

Instructor:
Prof. Selman
selman@cs.cornell.edu

Overview & Introduction

(Readings R&N: Chapter 1)

http://www.cs.cornell.edu/courses/cs4700/2017fa/

Stuart
Russell
Peter
Norvig

Artificial Intelligence
A Modern Approach
Third Edition

AI Methodology

Theoretical aspects

- **Mathematical formalizations, properties, algorithms**

Engineering aspects

- **The act of building (useful) machines**

Empirical science

- **Experiments**

What's involved in Intelligence?

A) Ability to interact with the real world

to perceive, understand, and act

speech recognition and understanding (*natural language*)

image understanding (*computer vision*)

B) Reasoning and Planning

CS4700

modeling the external world

problem solving, planning, and decision making

ability to deal with unexpected problems, uncertainties

C) Learning and Adaptation

Lots of data. Use to train statistical models.

We are continuously learning and adapting.

We want systems that adapt to us!

AI leverages from different disciplines

philosophy

e.g., foundational issues (can a machine think?), issues of knowledge and believe, mutual knowledge

psychology and cognitive science

e.g., problem solving skills

neuro-science

e.g., brain architecture

computer science and engineering

e.g., complexity theory, algorithms, logic and inference, programming languages, and system building.

mathematics, statistics, and physics

e.g., statistical modeling, continuous mathematics, statistical physics, and complex systems.

Historical Perspective

Obtaining an understanding of the human mind is one of the final frontiers of modern science.

Founders:

George Boole, Gottlob Frege, and Alfred Tarski

- **formalizing the laws of human thought**

Alan Turing, John von Neumann, and Claude Shannon

- **thinking as computation**

John McCarthy (Stanford), Marvin Minsky (MIT), Herbert Simon and Allen Newell (CMU)

- **the start of the field of AI (1956)**

History of AI:
The gestation of AI 1943-1956
(See Russell & Norvig)

1943 McCulloch and Pitts

- **McCulloch and Pitts' model of artificial neurons**
- **Minsky's 40-neuron network**

1950 Turing's "Computing machinery and intelligence"

1950s Early AI programs, including Samuel's checkers program, Newell and Simon's Logic theorist

1956 Dartmouth meeting : Birth of "Artificial Intelligence"

- 2-month Dartmouth workshop; 10 attendees
- Name was chosen. AI

History of AI:

(1952-1969)

Early enthusiasm, great expectations

1957 Herb Simon (CMU):

It is not my aim to surprise or shock you – but the simplest way I can summarize is to say that <u>there are now in the world machines that think, that learn, and that create</u>. ☺

1958 John McCarthy's LISP (symbol processing at core)

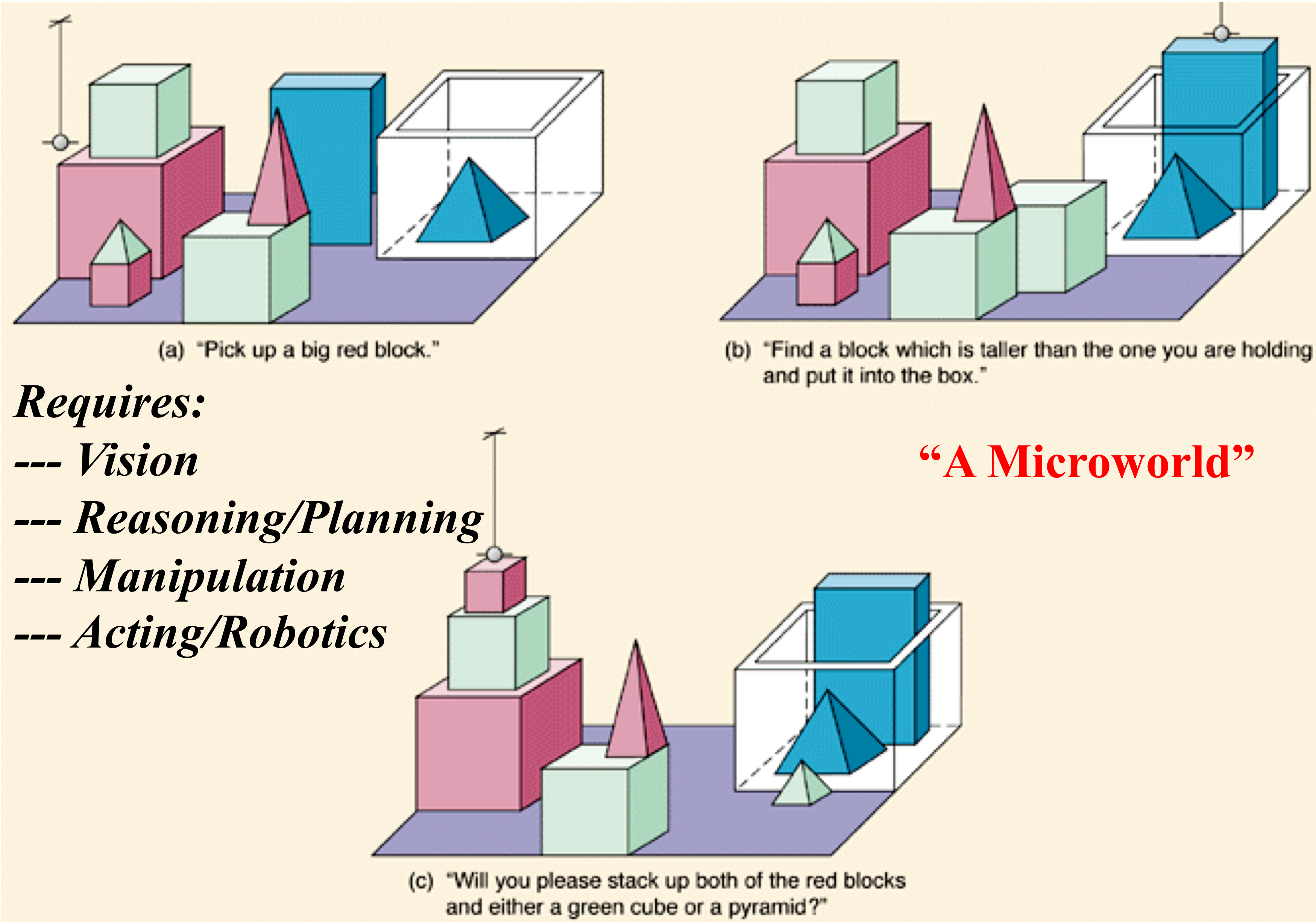
1965 J.A. Robinson invents the resolution principle, basis for automated theorem. General reasoning procedure.

Limited intelligent reasoning in microworlds

(such as the "blocks world" --- a toy robotics domain)

The Blocks World

gripper



(a) "Pick up a big red block."

(b) "Find a block which is taller than the one you are holding and put it into the box."

(c) "Will you please stack up both of the red blocks and either a green cube or a pyramid?"

Requires:

--- Vision

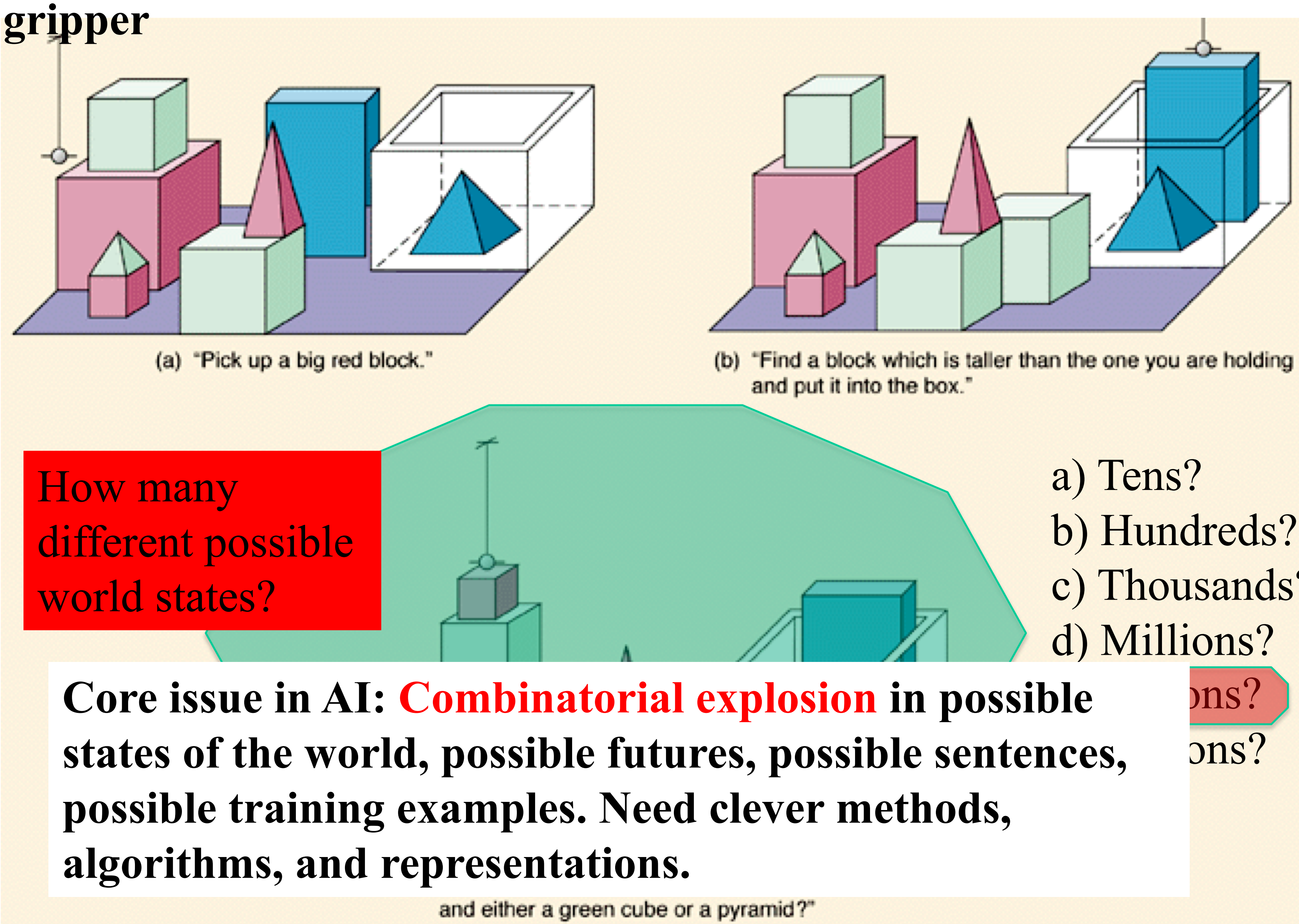
--- Reasoning/Planning

--- Manipulation

--- Acting/Robotics

"A Microworld"

Micro-world: The Blocks World



(a) "Pick up a big red block."

(b) "Find a block which is taller than the one you are holding and put it into the box."

How many different possible world states?

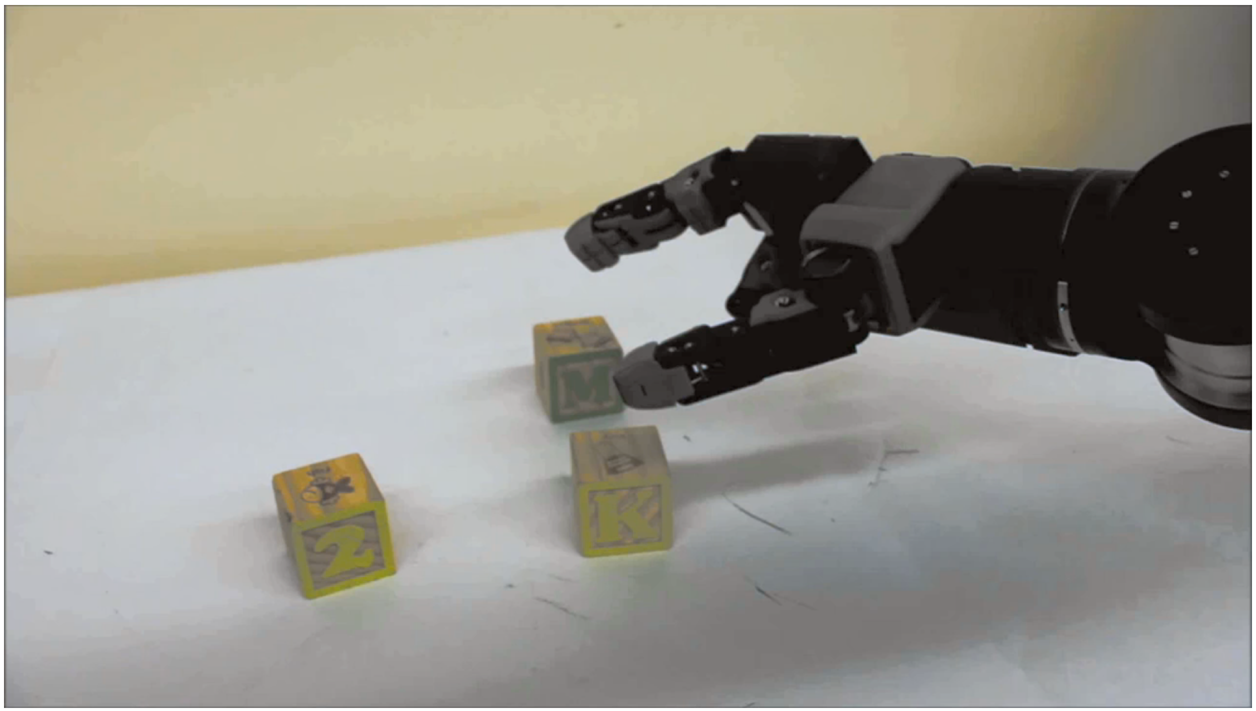
a) Tens?
b) Hundreds?
c) Thousands?
d) Millions?

Core issue in AI: Combinatorial explosion in possible states of the world, possible futures, possible sentences, possible training examples. Need clever methods, algorithms, and representations.

and either a green cube or a pyramid?"

"Brainy, Yes, but Far From Handy"
New York Times 09/01/14

Making dexterous hands with human-level touch and sensing still a real challenge. Link.



Stacking blocks may seem like an easy task for a human, but robots have long struggled with such fine control. HDT's Adroit manipulator uses **force-sensing and vision** to accomplish the delicate task.

Dynamic human touch — for example, when a finger slides across a surface — could distinguish ridges no higher than 13 nanometers, or about 0.0000005 of an inch. Individual molecules...

History of AI

A dose of reality (1965 - 1978)

1) Weizenbaum's ELIZA ("fools" users)

Capturing general knowledge is hard.

Revival: Amazon's Chatbots

2) Difficulties in automated translation

See Babelfish

Syntax and dictionaries are not enough

Consider going from English to Russian back to English.

Early effort…

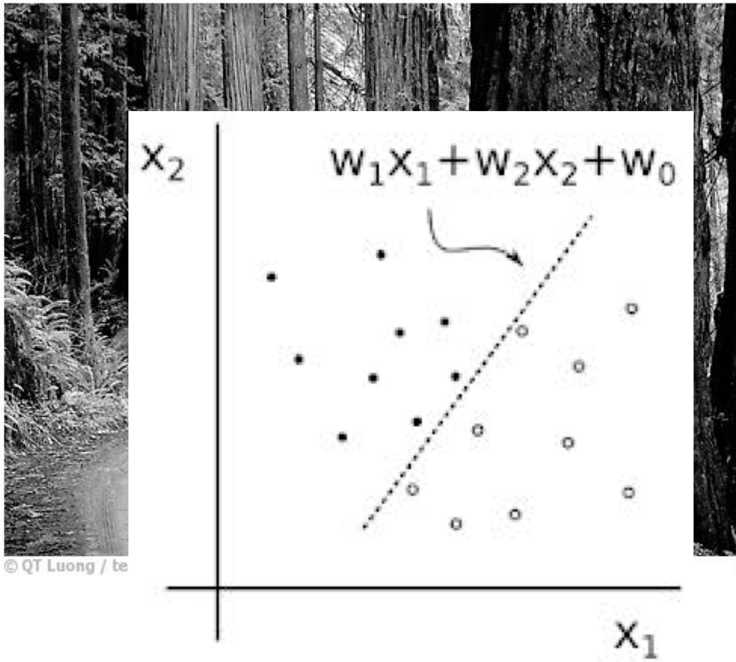
"The spirit is willing but the flesh is weak."

"The vodka is good but the meat is rotten."

Natural language processing (NLP) is hard.
(Ambiguity! Context! Anaphora resolution.)

History of AI

A dose of reality, cont. (1965 - 1978)



3) Cars climbing up trees (at CMU)…

Road sides look like parallel lines.

But, unfortunately, so do trees!

Computer vision is hard.
(Ambiguity! Context! Noisy pixels.)

4) Limitations of perceptrons discovered

Minsky and Papert (1969)

Can "only" represent linearly separable functions

Neural network research almost disappears

Machine learning is hard.

5) Intractability of inference. NP-Completeness (Cook 72)

Intractability of many problems attempted in AI.

Worst-case result….

Machine reasoning is hard.

History of AI

Knowledge based systems (1969-79)

Intelligence requires knowledge

Knowledge-based systems (lots of knowledge with limited but fast reasoning)

(Feigenbaum)

versus

general "weak" methods (a few basic principles with general reasoning)

(Simon and Newell)

Some success: *Expert Systems*

- Mycin: diagnose blood infections (medical domain)
- R1 : configuring computer systems
- AT&T phone switch configuration

Knowledge in rules of form:

If sympton_1 & sympton_3 then disease_2 (with certainty .8)

Surprising insight: Modeling medical expert easier than modeling language / vision / reasoning of 3 year old. (not foreseen)

Expert Systems

Very expensive to code. ($1M+)

Response: Try to learn knowledge from data.

Weak with uncertain inputs / noisy data / partial information

Response: Incorporate probabilistic reasoning

Brittle! (fail drastically outside domain)

But IBM's Watson's knowledge modules have expert system flavor!

Leads to 1980 -- 1995:

--- General foundations reconsidered

--- Foundations of machine learning established (e.g. computational learning theory; PAC learning; statistical learning)

--- Foundations of probabilistic formalisms: Bayesian reasoning; graphical models; mixed logical and probabilistic formalisms.

From 1995 onward:

--- Data revolution combined with statistical methods

--- Building actual systems

--- Human world expert performance matched (and exceeded) in certain domains

History of AI: 1995 - present

Several success stories with high impact …

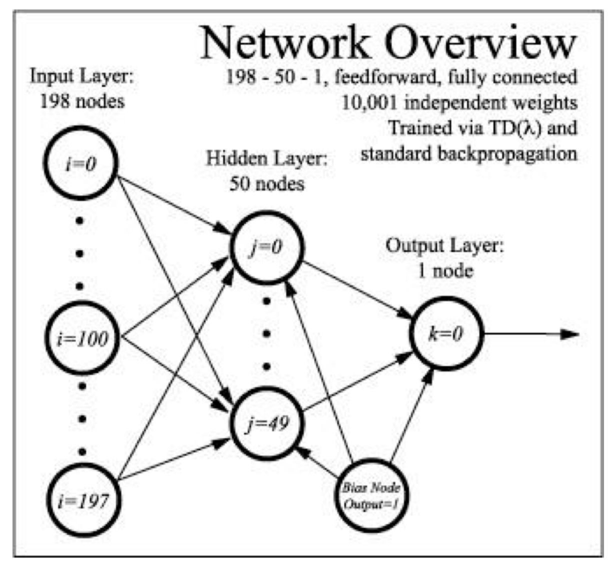
Machine Learning

In ’95, TD-Gammon.

World-champion level play by Neural Network that learned from scratch by playing millions and millions of *games against itself!* (about 4 months of training. Temporal-Difference learning.)

(initial games hundreds of moves)

Has changed human play.



Remaining open question: Why does this NOT work for, e.g., chess??

Some further remarks on reinforcement learning

--- reinforcement learning

strengthen behavior with positive reward

weaken with negative reward (punishment)

--- 2 versions of same program playing against itself.

give more detail on output: one possible architecture

game state and who is on play is input

Then, for each possible move:

NN computes score/float y in range <-1,+1>.

--- After win (or loss), adjust weights in gradient descent direction to move score y for that move up (or down). for that move.

So, let’s say network made move “A” in play state S at because move A received a score of 0.9 by the neural net for state S..

Then, after playing out the game, the network lost. So, more likely than not, move A was quite possibly not the right move to make in state S.

Therefore, adjust weights on network a tiny bit to move down for move A, given game is in state S.

Score(state S) is a function of the state and the setting of the weights in the NN. Use basic calculus (gradient descent), to change the weights to lower (or raise) the score given the inputs representing state S.

The essence of NN is that it reduces a lot of AI/ML
to gradient descent optimization, given some training
set and a loss function. The loss function specifies what
we want the NN to compute.

So, the NN is going to compute a complex function
Given the input state and the setting of its weights.

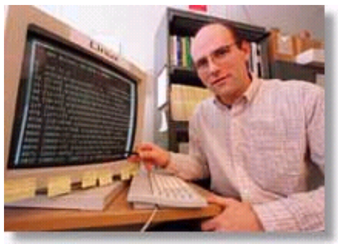
Learning / behavior etc., all comes down to modifying
the weights to have the NN output something as close
as possible to the training example points of the
behavior/function that needs to be learned.

In a game, for each possible move, you want to know the minimax value
or a good approximation thereof ("probability" first player wins between
-1.0 (certain
loss and +1.0 (certain win)).

Lots of AI/ML reduced to cleverly designed optimization problems.

1996 --- EQP:
"Robbin's Algebras are all Boolean"

A mathematical conjecture (Robbins conjecture) unsolved for 60 years!



First creative mathematical proof by computer.
Contrast with brute-force based proofs such as the 4-color theorem.

The Robbins problem was to determine whether one particular set of rules is powerful enough to capture all of the laws of Boolean algebra.

Mathematically:
Can the equation not(not(P))= P be derived from the following three equations?
[1] (P or Q) = (Q or P)
[2] (P or Q) or R = P or (Q or R),
[3] not(not(P or Q) or not(P or not(Q))) = P.

[An Argonne lab program] has come up with a major mathematical proof that would have been called creative if a human had thought of it.
New York Times, December, 1996

http://www-unix.mcs.anl.gov/~mccune/papers/robbins/

ROBBINS CONJECTURE

THE PROOF ?

7	$\overline{\overline{p+q}+\overline{\bar{p}+q}}=q$	[*Robbins axiom*]
10	$\overline{\overline{\overline{p+q}+\overline{\bar{p}+q}}+q}=\overline{p+q}$	[7 → 7]
11	$\overline{\overline{\overline{\bar{p}+q}+\overline{p+q}}+q}=\overline{\bar{p}+q}$	[7 → 7]
29	$\overline{\overline{\overline{\bar{p}+q}+p+2q}+\overline{\bar{p}+q}}=q$	[11 → 7]
8865	$\overline{\overline{\overline{3p}+p}+\overline{3p}+2p+\overline{3p}}=\overline{\overline{3p}+p}+2p$	[8855 → 7]
8866	$\overline{\overline{3p}+p+\overline{3p}}=p$	[8855 → 7, *simp* : 11]
8870	$\overline{\overline{\overline{3p}+p+\overline{3p}}+q+\overline{p+q}}=q$	[8866 → 7]
8871	$\overline{\overline{3p}+p}+2p=2p$	[8865, *simp* : 8870]

A Baker's Dozen. *The key steps in proving the Robbins conjecture, as reported by EQP, an automated theorem-proving program developed by William McCune and colleagues at Argonne National Laboratory. (See Box, "Substitute Teacher," page 63 for details.)*

As Eas

The phrase *il est* repeatedly in the French mathematician phrase is common in spelling out details th matical techniques. words. What Laplac intense mathematical Oddly enough, La see how a particular ...sy to see. Th ...mbination of ...ld a computer ...iam McCune, ...tory in Illinoi ...g programs. ...n a range of pr ...ost powerful pr ... every bit as ...m considers e ...gram's creato ...ptic or not, E announced a solutio bolic logic that ha posed in the 1930's. has been solved by described as reasoni "It's a clear lan Stanley Burris, a lo Canada. "Now tha problem, it opens th The Robbins co which are named (1815–1864), who into algebraic expr expressed as $p+$ $p+N(q)$; a logic

New: 2014 ---
Erdos Discrepancy Conjecture resolved

Note: Same order of search complexity as performed by Deep Blue per move. Quantative threshold for creativity?

Erdos Discrepancy Conjecture

A recently resolved math challenging problem using automated reasoning.

A conjecture about properties of infinite sequences of +1s and -1s.

Let’s cover some details.

Example Consider a sequence of 1s and -1s, e.g.: \sigma x_i

-1,	1,	1,	-1,	1,	1,	-1,	1,	-1	…
1	2	3	4	5	6	7	8	9	…
	2		4		6		8		…
		3			6			9	…

and look at the sum of sequences and subsequences:

-1 + 1 = 0 (x_i)

-1 + 1 + 1 = 1

-1 + 1 + 1 + -1 = 0

-1 + 1 + 1 + -1 + 1 = 1

-1 + 1 + 1 + -1 + 1 + 1 = 2

-1 + 1 + 1 + -1 + 1 + 1 + -1 = 1

-1 + 1 + 1 + -1 + 1 + 1 + -1 + 1 = 2

-1 + 1 + 1 + -1 + 1 + 1 + -1 + 1 + - 1 = 1

etc.

and "skip by 1" (x_2i)

1 + -1 = 0

1 + -1 + 1 = 1

1 + -1 + 1 + 1 = 2

etc.

and "skip by 2" (x_3i)

1 + 1 = 2

1 + 1 + -1 = 1

etc.

What happens to partial sums?

How would you tackle the problem as a Computer Scientist?

What is the size of the search space? How long would your algorithm take?

Discovered in 2015: there exists a sequence of *1160* +1s and -1s such that sums of all subsequences *never* < -2 or > +2.

Result was obtained with a *general* reasoning program (a Boolean Satisfiability or SAT solver). *Surprisingly*, the approach far outperformed specialized search methods written for the problem, including ones based on other known types of sequences. (A PolyMath project started in January 2010.)

−	+	+	−	+	−	−	+	+	−	+	+	−	+	−	−	+	−	−	+	+	−	+	−	−	+	−	−	+
+	−	+	−	−	+	+	−	+	+	−	+	−	+	+	−	−	+	+	−	+	−	−	−	+	−	+	+	−
+	−	−	+	−	−	+	+	+	+	−	−	+	−	−	+	+	−	+	−	−	+	+	−	+	+	−	−	−
−	+	+	−	+	+	−	+	−	+	+	−	−	+	+	−	+	−	+	−	−	−	+	+	−	+	−	−	+
+	−	+	+	−	+	−	−	+	+	−	+	−	−	+	−	−	−	+	−	+	+	−	+	−	−	+	+	−
+	+	−	+	−	−	+	−	−	+	+	−	+	+	−	+	−	−	+	+	−	+	−	−	+	+	+	−	+
−	+	−	−	−	−	+	+	+	−	+	−	−	+	−	−	+	+	+	−	−	−	+	+	−	+	+	−	+
−	−	+	−	−	+	+	+	−	−	+	−	+	−	+	−	−	+	−	+	+	+	−	+	+	−	+	−	−
+	−	−	+	+	−	+	−	−	+	+	−	+	+	−	+	−	−	+	−	−	+	+	−	−	+	+	+	−
−	−	+	+	+	−	+	−	−	−	+	+	−	+	−	−	+	+	−	−	+	−	+	−	−	+	−	+	+
+	−	+	−	−	+	+	−	+	+	−	+	−	−	+	+	−	+	−	−	+	−	−	+	+	−	+	−	−
+	+	−	−	+	−	+	+	−	+	−	+	−	−	+	−	+	−	+	+	−	+	−	−	+	+	−	+	−
−	+	−	−	+	+	−	+	−	+	−	+	+	−	+	−	+	−	+	+	−	−	−	+	−	+	−	−	+
+	+	+	−	−	+	−	−	−	+	+	−	+	−	+	+	−	+	−	−	+	+	−	+	−	−	+	−	−
+	+	−	+	−	−	+	+	+	+	−	−	+	−	−	−	+	−	+	+	+	+	−	−	+	−	−	+	+
−	+	+	−	+	−	−	+	+	−	+	−	−	+	−	−	+	+	−	+	−	−	+	+	−	+	+	−	+
−	−	+	+	−	+	−	−	+	−	−	+	+	−	+	+	−	+	−	−	−	−	+	+	+	−	+	−	−
+	+	−	−	+	+	+	−	−	−	+	−	+	+	−	+	−	−	+	−	+	+	−	−	−	+	−	+	+
−	+	+	−	+	−	−	+	−	−	+	+	−	−	+	+	+	+	−	+	−	−	+	−	−	+	−	−	+
+	+	+	−	−	+	−	−	+	+	+	−	−	−	+	+	−	+	+	−	+	−	−	+	+	−	−	+	−
+	−	−	+	−	−	+	+	−	+	+	−	+	−	−	+	−	−	+	−	+	+	+	−	+	+	−	+	−
−	+	−	−	+	+	−	−	+	−	+	+	−	+	+	−	+	−	−	+	−	−	+	−	−	+	+	−	+
+	−	+	−	−	+	+	+	−	−	−	+	+	−	+	−	−	+	+	−	+	+	−	−	−	+	+	+	−
−	−	+	+	−	+	+	−	−	−	−	+	+	+	−	−	+	−	+	+	−	+	−	−	+	−	−	+	+
−	+	−	−	+	+	−	+	+	−	+	−	+	+	−	−	+	+	−	−	+	+	−	−	−	−	+	+	+
−	+	+	−	−	+	+	−	−	−	−	+	+	−	+	+	+	−	−	+	+	−	−	−	+	+	+	−	−
−	−	+	−	+	−	+	+	−	+	+	−	+	+	−	+	−	+	−	−	−	−	+	+	+	−	−	+	+
−	+	−	−	+	+	−	+	+	−	+	−	−	+	−	−	+	−	−	+	+	−	+	−	−	+	+	−	+
+	−	+	−	−	+	+	−	−	+	−	+	−	−	+	−	+	−	+	−	+	+	+	+	−	−	−	+	−
+	−	+	+	−	−	+	−	−	+	−	+	−	+	−	+	+	−	+	−	+	+	+	−	−	+	−	+	−
−	+	−	−	+	−	+	+	+	−	−	+	−	+	+	+	−	−	−	+	+	−	+	−	−	+	−	−	+
+	−	+	+	−	−	+	+	−	−	−	+	+	−	+	−	+	+	−	−	+	+	−	+	−	−	−	+	−
+	+	−	+	−	−	+	−	+	+	−	−	+	+	−	+	−	−	+	+	−	+	−	−	+	−	+	+	+
−	+	−	−	+	+	−	−	+	−	+	−	+	+	+	−	−	+	−	+	−	−	+	+	−	+	+	−	+
−	−	+	−	−	+	−	+	+	−	−	−	+	−	+	+	−	+	−	+	+	−	−	+	+	−	+	−	−
+	+	+	−	+	−	−	−	−	+	+	−	−	+	−	+	+	−	+	−	+	+	−	−	+	+	−	+	−
−	+	+	−	+	−	+	+	−	−	+	+	−	+	−	−	−	+	−	+	+	−	+	−	−	+	+	+	−
−	−	−	+	−	+	−	+	+	−	−	+	+	−	+	−	−	+	+	−	+	+	−	+	+	−	+	−	−
+	−	−	+	−	−	+	+	+	+	−	−	−	+	+	−	−	−	+	−	+	−	+	+	−	+	−	+	+
+	−	−	+	−	+	+	−	−	+	−	+	−	−	+	−	+	−	+	+	−	−	−	+	+	+	−	+	+

1160

elements

all sub-sums

stay between

-2 and +2

40 x 29 pattern

Aside: A Taste of Problem Size

Consider a real world Boolean Satisfiability (SAT) problem, from software & hardware verification.

The instance bmc-ibm-6.cnf, IBM LSU 1997:

Each line gives a brief logical statement

("0" marks end of line)

```
p cnf
-1 7 0
-1 6 0
-1 5 0
-1 -4 0
-1 3 0
-1 2 0
-1 -8 0
-9 15 0
-9 14 0
-9 13 0
-9 -12 0
-9 11 0
-9 10 0
-9 -16 0
-17 23 0
-17 22 0
```

"1" for variable x_1, "2" for x_2, etc.

x_1, x_2, x_3, … our Boolean variables (set to True or False)

((not x_1) or x_7)

((not x_1) or x_6)

etc.

Question: Can we satisfy all statements?

Set x_1 to False ??

SAT problem lies at the core of computer science

Prototypical NP-complete problem (from P vs. NP)

10 pages later:

185 −9 0
185 −1 0
177 169 161 153 145 137 129 121 113 105 97
89 81 73 65 57 49 41
33 25 17 9 1 −185 0
186 −187 0
186 −188 0
…

I.e., (x_177 or x_169 or x_161 or x_153 …
x_33 or x_25 or x_17 or x_9 or x_1 or (not x_185))

clauses / constraints are getting more interesting…

Note x_1 …

4000 pages later:

10236 −10050 0
10236 −10051 0
10236 −10235 0
10008 10009 10010 10011 10012 10013 10014
10015 10016 10017 10018 10019 10020 10021
10022 10023 10024 10025 10026 10027 10028
10029 10030 10031 10032 10033 10034 10035
10036 10037 10086 10087 10088 10089 10090
10091 10092 10093 10094 10095 10096 10097
10098 10099 10100 10101 10102 10103 10104
10105 10106 10107 10108 −55 −54 53 −52 −51 50
10047 10048 10049 10050 10051 10235 −10236 0
10237 −10008 0
10237 −10009 0
10237 −10010 0

…

Finally, 15,000 pages later:

−7 260 0
7 −260 0
1072 1070 0
−15 −14 −13 −12 −11 −10 0
−15 −14 −13 −12 −11 10 0
−15 −14 −13 −12 11 −10 0
−15 −14 −13 −12 11 10 0
−7 −6 −5 −4 −3 −2 0
−7 −6 −5 −4 −3 2 0
−7 −6 −5 −4 3 −2 0
−7 −6 −5 −4 3 2 0
185 0

Search space of truth assignments:

$$2^{50000} \approx 3.160699437 \cdot 10^{15051}$$

Current reasoning engines can solve this instance in a few seconds! (no satisfying assignment exists + proof)

Reminder: Consider a sequence of 1s and -1s, e.g.:

-1,	1,	1,	-1,	1,	1,	-1,	1,	-1	…
1	2	3	4	5	6	7	8	9	…
	2		4		6		8		…
		3			6			9	…

and look at the sum of sequences and subsequences:

-1 + 1 = 0

-1 + 1 + 1 = 1

-1 + 1 + 1 + -1 = 0

-1 + 1 + 1 + -1 + 1 = 1

-1 + 1 + 1 + -1 + 1 + 1 = 2

-1 + 1 + 1 + -1 + 1 + 1 + -1 = 1

-1 + 1 + 1 + -1 + 1 + 1 + -1 + 1 = 2

-1 + 1 + 1 + -1 + 1 + 1 + -1 + 1 + - 1 = 1

etc.

and "skip by 1"

1 + -1 = 0

1 + -1 + 1 = 1

1 + -1 + 1 + 1 = 2

etc.

and "skip by 2"

1 + 1 = 2

1 + 1 + -1 = 1

etc.

Back to sequences of +1/-1s

Logic / SAT encoding has variables for the sequence X_1, X_2, …, X_N

(we interpret True for +1 and False for -1)

but also e.g.

Proposition: "sum_of_first_2_terms_of_skip_by_2_subseq_=_2"

(for any given setting of X_1 … X_N this is either True or False)

and statements of the form:

IF ((sum_of_first_2_terms_of_skip_by_3_subseq_=_2 == True)

AND (X_9 == False))

THEN

(sum_of_first_3_terms_of_skip_by_3_subseq_=_1 == True)

Why? **SAT form?**

Encoding: 37,418 variables and 161,460 clauses / constraints.

Sequence found in about 1 hour (MacBook Air).

Perhaps SAT solver was "lucky" in finding the sequence?

Remarkably, SAT solver also shows that each sequence of *1161* or longer leads to +3 (or -3) somewhere. (Erdos discrepancy conjecture)

(Again, think of the size of the search space!)

Encoding: 37,462 variables and 161,644 clauses / constraints.

Proof of non-existence of discrepancy 2 sequence found in about 10 hour (MacBook Air).

Proof: 13 gigabytes and independently verified (50 line proof checking program). Proof is around a billion small inference steps.

Machine understands and can verify result easily (milliseconds); Humans: probably never. Still, we can be certain of the result because of the verifier.

Observations

1) **Result different from earlier "computer math" results, such as the proof of the 4 color theorem, because here we don't need to trust the theorem prover. Final proof ("certificate") can be checked easily by anyone.**

2) **It's *not* a brute force search. Earlier SAT solvers cannot find the proof. Specialized programs cannot find the proof.**

   Brute force proof is of order 2^1161 = 3.13 x 10^349. Current solver finds complete proof with "only" around 1.2 x 10^10 steps. Clever learning and reasoning enables a factor 10^339 reduction in proof size.

3) **In part inspired by discrepancy 2 result, Terence Tao proved several months later the general Erdos conjecture (for any discrepancy). Deep and subtle math.**

4) **But, does not fully supersedes the 1161 result for the discrepancy 2. Future math may build further on these types of computational results. (I.e. true, verifiable facts but not human accessible.)**

1997:
Deep Blue beats the World Chess Champion

Deep Blue had Kasparov in deep thought
(CNN)

vs.

I could feel human-level intelligence across the room

Gary Kasparov, World Chess Champion (human…)

Note: when training in self-play, be careful to randomize!

Deep Blue vs. Kasparov

Game 1: 5/3/97:
Kasparov wins

Game 2: 5/4/97:
Deep Blue wins

Game 3: 5/6/97:
Draw

Game 4: 5/7/97:
Draw

Game 5: 5/10/97:
Draw

Game 6: 5/11/97:
Deep Blue wins

Game 3: Why did Kasparov not simply repeat moves from game 1?

The value of IBM's stock increased by $18 Billion!

We'll discuss Deep Blue's architecture, when we cover *multi-agent search*.

On Game 2

Game 2 - Deep Blue took an early lead. Kasparov resigned, but it turned out he could have forced a draw by perpetual check.

Interestingly, if Kasparov had been playing a human he would most likely not have resigned!

This was real chess. This was a game any human grandmaster would have been proud of.

Joel Benjamin

grandmaster, member Deep Blue team

Kasparov on Deep Blue

1996: Kasparov Beats Deep Blue

"I could feel --- I could smell --- a new kind of intelligence across the table." (CNN)

1997: Deep Blue Beats Kasparov

"Deep Blue hasn't proven anything." ☺

Current strongest play: Computer-Human hybrid

May, '97 --- Deep Blue vs. Kasparov. First match won against world-champion. ``intelligent creative'' play.

200 million board positions per second!

Kasparov: ... still understood 99.9 of Deep Blue's moves.

Deep Blue considers *60 billion boards per move*! Human?

Around 10 to 20 lines of play. Hmm…

Intriguing issue: How does human cognition deal with the search space explosion of chess?

Or how can humans compete with computers at all?? (What does human cognition do? Truly unknown…)

Concepts (briefly)
(more details with multi-agent search)

--- Minimax search on game tree to get optimal move (large tree >= 10^80 chess)

Size tree: b^d (b --- average branching; d --- depth)

alpha-beta pruning: b^(d/2) [key technique]

--- Board evaluation or utility function when

you can’t search to the bottom

--- Board eval is linear weighted some of features; can be trained via learning. (Reinforcement learning / AlphaGo)

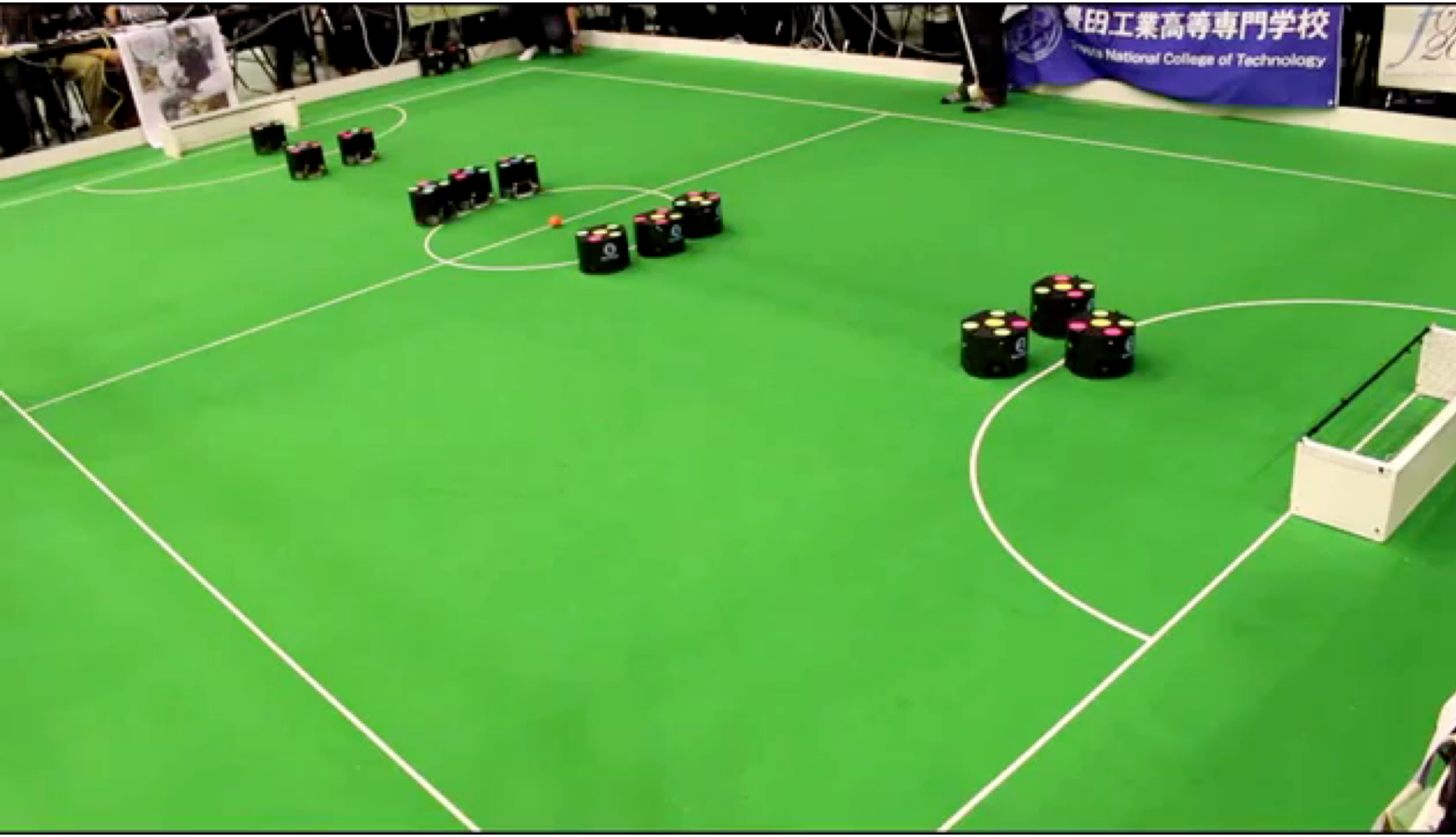
--- Chess complexity?

O(1) (formally speaking…)

--- 2017: AlphaGo beats world human Go champion

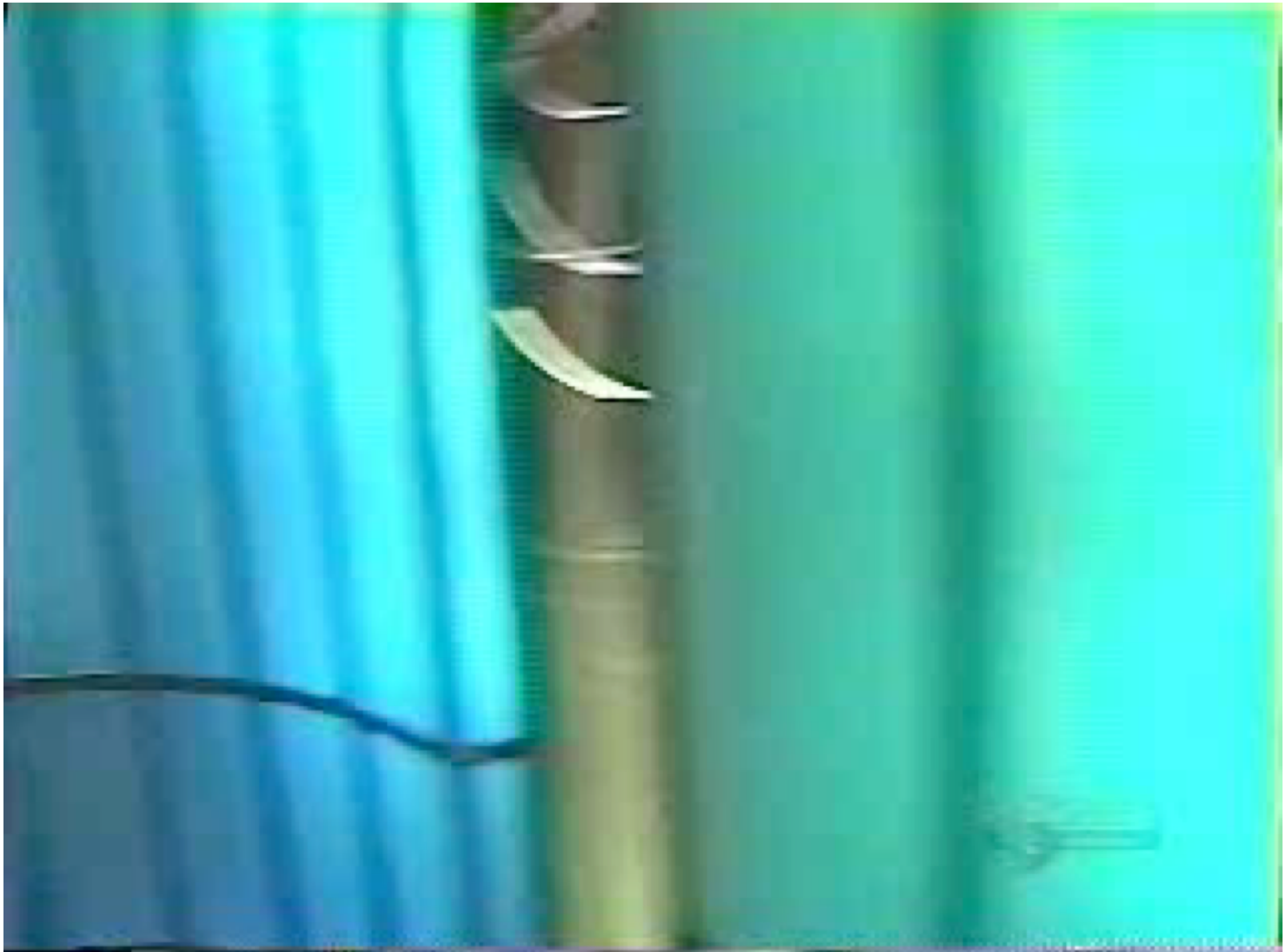
Robocup @ Cornell --- Raff D’Andrea 2000

RoboCup Japan open 2013



Kiva Systems $700M

From Robocup to Warehouse Automation



2005 Autonomous Control: DARPA GRAND CHALLENGE

October 9, 2005

Stanley and the Stanford RacingTeam were awarded 2 million dollars for being the first team to complete the 132 mile DARPA Grand Challenge course (Mojave Desert). Stanley finished in just under 6 hours 54 minutes and averaged over 19 miles per hours on the course.

Cornell team stuck ☹
due to malfunctioning GPS.

Sebastian Thrun:
Google's driverless car (2011)

http://www.youtube.com/watch?v=bp9KBrH8H04 cornell darpa grand challenge

People | Technology | Media | Sponsors | Contact Us | Internal

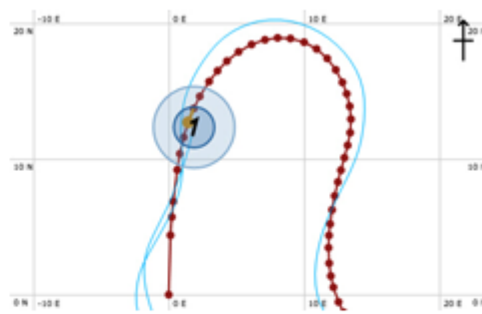
Pick a tab to navigate

path **planning**

A* algorithm Covered in search and problem solving.

Path Planning Overview
Path planning is the basic process by which our vehicle decides on what path to take through the world. The A.I. uses the world model created by the sensors, the GPS waypoints provided during the race by DARPA, and a road following algorithm to pick a best path.

Road Following
The road following algorithm uses color differences, shadowing, and edge-detection algorithms to detect the sides of a road (if there is a road) and then decides if the road is turning, going straight, which direction, how sharply, etc. The road following algorithm uses input from most of the vehicle sensors, and provides the A.I. with probable road characteristics.



The red line represents the ideal path picked by our A.I., and the turquoise path represents the actual path traveled by our vehicle. The differences arise because 1) we did not start our vehicle on the ideal path, and 2) our vehicle must, without exceeding its performance limits, avoid small obstacles such as boulders.

Cornell: 4th!
Also, in historic
***1st autonomous driverless car collision*. Rear-ended by MIT car!**

2007 Darpa Urban Challenge
Winner: CMU Tartan Racing's Boss

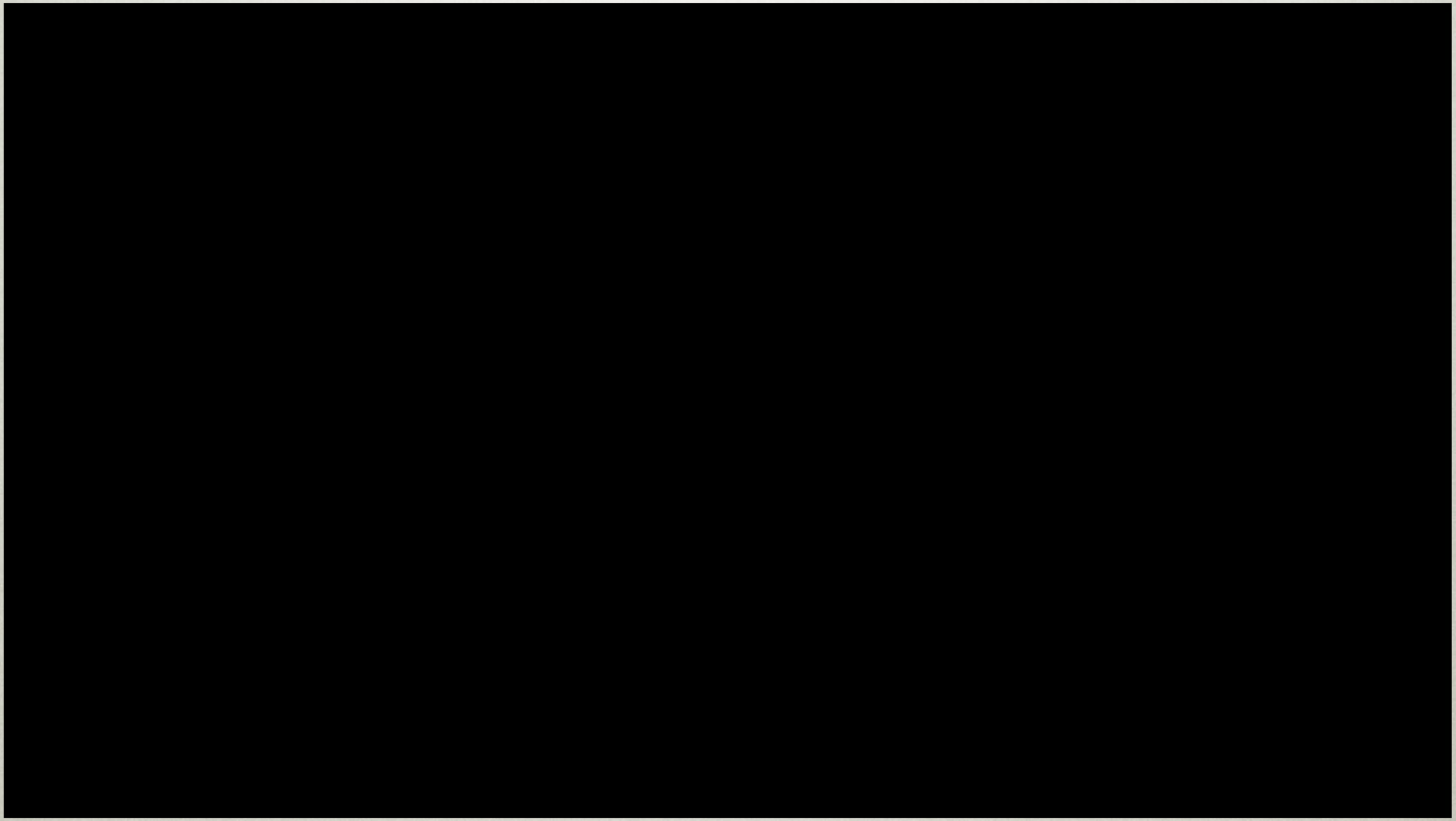
http://www.tartanracing.org/blog/index.html#26

Watson: Question-Answering system, 2011

Watson defeats the
two greatest Jeopardy!
champions

http://www.youtube.com/watch?v=dr7IxQeXr7g

Watson



Neural Networks --- Deep Learning, 2012.

New York Times: "Scientists See Promise in Deep-Learning Programs," Saturday, Nov. 24, 2012.

http://www.nytimes.com/2012/11/24/science/scientists-see-advances-in-deep-learning-a-part-of-artificial-intelligence.html?hpw

Multi-layer neural networks, a resurgence!

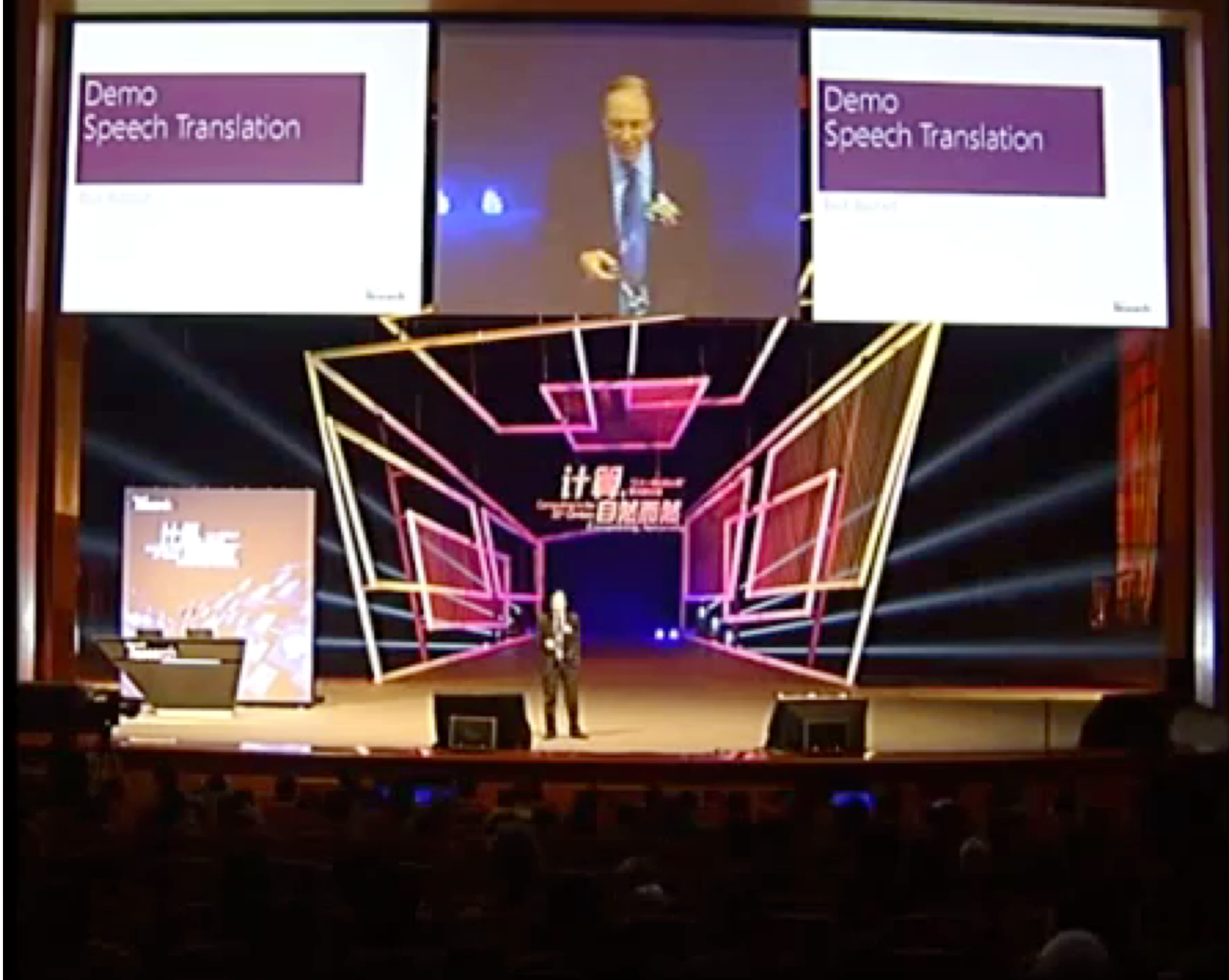
a) Winner one of the most recent learning competitions

b) Automatic (unsupervised) learning of "cat" and "human face" from 10 million of Google images; 16,000 cores 3 days; multi-layer neural network (Stanford & Google).

c) Speech recognition and real-time translation (Microsoft Research, China).

Aside: see web site for great survey article "A Few Useful Things to Know About Machine Learning" by Domingos, CACM, 2012.

ML as Optimization (i.e. minimize a loss function)



Start at min. 3:00. Deep Neural Nets in speech recognition.

Other promising ongoing efforts

1) Intelligent autonomous assistants, e.g., iPhone's Siri (still a long way to go ☺) Integrated, autonomous agents. Google Glass will be the next step. Location / context aware; rich sensing, vision and speech understanding and generation.

2) Fully self-driving car (Google; assisted driving Mercedes and BMW --- the cost of a car is becoming software and sensors Incredibly more lines of code in a Mercedes than in a Boeing 747.)

2) Google translate. Reaches around 70% of human translator performance. Almost fully a purely statistical approach.

Not clear yet how far one can go without a real understanding of the semantics (meaning). But with Big Data, statistical methods already went much further than many researchers had considered possible only 10 years ago.

Course Administration ✓

What is Artificial Intelligence? ✓

Course Themes, Goals, and Syllabus

Setting expectations for this course

Are you going to build real systems and robots?

NO…

Goal:
Introduce the conceptual framework and computational techniques that serve as a foundation for the field of artificial intelligence (AI).

Syllabus

- Structure of intelligent agents and environments.
- Problem solving by search: principles of search, uninformed ("blind") search, informed ("heuristic") search, and local search.
- Constraint satisfaction problems: definition, search and inference, and study of structure.
- Adversarial search: games, optimal strategies, imperfect, real-time decisions.
- Logical agents: propositional and first order logic, knowledge bases and inference.
- Uncertainty and probabilistic reasoning: probability concepts, Bayesian networks, probabilistic reasoning over time, and decision making.
- Learning: inductive learning, concept formation, decision tree learning, statistical approaches, neural networks, reinforcement learning.

So far, we discussed

Artificial Intelligence and characteristics of *intelligent* systems.

Brief history of AI

Major recent AI achievements

Reading: Chapter 1 Russell & Norvig