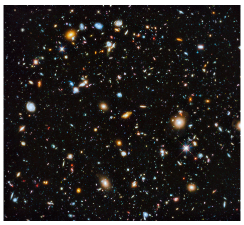
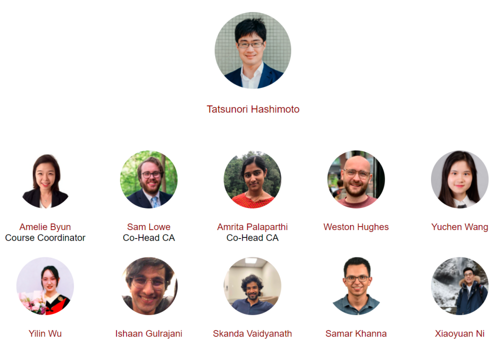
General: Overview



The instruction team

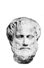


- With us are going to be a team of experienced course assistants
- (intros)
- I know that you all come from a wide range of backgrounds
- Some of you are taking your first AI course
- Others of you already have substantial AI or machine learning experience
- Regardless of your background, we are here to help you out
- If you have a tough time with some background material – come to us
- If you find the course basic, and want to talk about more advanced topics in AI – come to us!
- This is the first time in two years that this course is returning to an in-person format
- there are logistical changes to the 221 format so
- we will go over the changes next

What's next?

 Course logistics

 Course content

 AI history

 AI today

- We will be going over the course structure and logistics first
- Then, as a quick teaser, give an overview of the material
- Finally, we'll talk through the history of AI, and how we got to where we are today

Activities

Lecture	Mon/Wed	90 min lecture + QA (recorded + zoom option)
Problem solving section	Thu	CAs work out practice problems + QA
Open office hours	Hybrid	OH with instructors and CAs
Exam	Gradescope	Final, 3 hours

All the details on the website (cs221.stanford.edu)

- The format is fairly standard
- There is lectures, sections, and office hours
- These will be streamed on zoom and recorded for SCPD
- On thursdays, we will have **problem sessions**, where CAs will walk you through practice problems,
- Due to some SCPD-compatible classroom shortages, we are finalzing the time and location, but it will be announced on Ed as soon as we settle on this
- This is like a traditional section that will prepare you for exams.
- The instructors and CAs will have **office hours** to help with questions about the course or homeworks. Please see the website for the specifics
- Outside of these recurring events, there will be a final exam which is 3 hours.
- Finally, you can find all of these details on the website.

Prerequisites

Programming (Python)	CS 106A, CS 106B, CS 107
Discrete math, mathematical rigor	CS 103
Probability	CS 109
Basic linear algebra	Math 51

- More about general familiarity rather than specific knowledge
- Prerequisite modules provide refresher
- Foundations homework will give you an idea of what to expect

- This course requires a strong foundation in a number of areas.
- You need to be able to program, ideally in Python.
- You should comfortable with mathematical notation
- As in the title, we will cover the foundations of AI, and that involves discrete math
- And finally, you should know basic probability and linear algebra to understand machine learning.
- I would emphasize that it's less important that you know particular things
- (e.g., we don't use eigenvectors in this course even though that's a pillar of any linear algebra course),
- While it is possible to fill in the gaps, this course does move quickly,
- Ideally you want to be focusing your energy on learning AI rather than catching up on prerequisites.
- If you need a refresher, we have links to the refresher materials from past offerings of 221 on the website
- If you're unsure about the prerequisites, the first homework will test some of this.

Grading

Homeworks	75%
Final	25%
Project	(extra credit)
Piazza contributions	(extra credit)

Letter Grade or Satisfactory / No Credit

- Your grade will consist of homeworks and exams, which will be worth 75 and 25%, respectively. This will be a bit of a change from the past, and we are increasing the relative contribution of the homeworks as this is where you will be spending the majority of your time in the course
- In addition, there will be a small amount of extra credit from doing projects or being helpful on piazza
- You may choose a letter grade or a satisfactory / no credit grading basis.

Homeworks

- 8 homeworks, mix of written and programming problems

Introduction	foundations
Machine learning	sentiment classification
Search	text reconstruction
MDPs	blackjack
Games	Pac-Man (+ competition with extra credit)
CSPs	course scheduling
Bayesian networks	car tracking
Logic	language and logic

- Code will be autograded, feedback on subset of test cases
- Autograding is strict, so double check your outputs carefully!
- 7 total late days, max two per homework
- 2nd homework has a longer deadline

- We will have 8 homeworks, which include a mix of written and programming parts,
- All the programming parts will be autograded when you submit on Gradescope.
- We will run your code on all test cases, but you will only get immediate feedback on a subset of the test cases.
- Your grade will be based on correctness on both the public and the hidden test cases.
- Autograding will be strict, so please make sure that you double check your outputs.
- Finally, you will have 7 late days to use across all the homeworks,
- Only at most two per homework,
- This is so that we can release homework solutions in a timely manner.

Project (optional)

- Goal: choose any task you care about and apply techniques from class
- Work in groups of up to 4
- Milestones: project interest form, proposal, progress report, video, final report
- Task is completely open, but must follow well-defined steps: task definition, implement baselines/oracles, evaluate on dataset, literature review, error analysis (read website)
- Help: assigned a CA mentor, come to any office hours

- This quarter the project will once again be optional
- Traditionally, the project has been a great way for students to demonstrate matery on a topic they care about
- This is a great way to get started in AI research, and Id encourage you to do the project if you have the time
- We will assign you a CA mentor who can guide you through the process,
- I also would encourage you to come to the instructors' office hours to discuss your ideas.

Exam

- 3-hour final exam on all course material
- Emphasis on the conceptual and mathematical contents
- Materials will closely follow lecture and section
- Important!: Please check and make sure you do not have exam time conflicts

- The other major evaluation for the course will be an exam
- The homeworks tests the hands-on, engineering aspects of AI and the exams test the mathematical and conceptual ones
- Coverage will be for the entire course, and the exam will be calibrated to take 3 hours maximum
- Unlike the homework, the exam will closely follow problems and facts that we present in lecture and review in section
- Finally - and this is important - please make sure you do not have conflicts with the stated exam times!

THE HONOR CODE

- Do collaborate and discuss together, but write up and code independently.
- Do not look at anyone else's writeup or code.
- Do not show anyone else your writeup or code or post it online (e.g., GitHub).
- When debugging with others, only look at input-output behavior.
- We will run MOSS periodically to detect code plagiarism.

- I want to talk about the honor code,
- 221 has been taught many times, so there's materials online
- This creates a temptation for cheating in various ways
- Don't do it! most likely we will end up finding it and have to report it
- Despite these warnings we still had to report 8 % of students last quarter
- While we do encourage you to collaborate,
- All your writeups and code should be done independently.
- You shouldn't be looking at anyone else's writeup including friends and old solutions
- We have automated ways of detecting some of this – dont think that the scale of the class makes it easy to cheat

Communication

- Public Ed post (course staff + students): ask general questions
- Private Ed post (course staff): ask questions that might give away answers
- Email cs221-spr2122-staff@lists.stanford.edu (Instructors + head TA + liaison): OAE and other sensitive matters
- Survey and feedback through the course (high resolution feedback)

- There are several ways to contact us,
- In general, we encourage you to use the most public venue that is appropriate,
- This will lead to faster response times and it will help others
- If you have any general questions about the course content or logistics, make a public Pizza post.
- If you have questions that might reveal solutions or anything about the quiz before it's due, make a private Piazza post.
- If you have any sensitive matters (e.g., OAE accommodations), you can email the four of us
- Finally, once in a while, we will conduct surveys periodically to see how things are going.

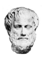
All details are on the course website (cs221.stanford.edu)

- This was alot to cover, but its all on the website!

What's next?

 Course logistics

 Course content

 AI history

 AI today

- Now we'll go through the major topics of the course, and how it relates to AI more broadly

Bridging the gap

- Important, real-world problems are often complex with lots of uncertainties.
- For example, we might want to build a system that can navigate through a busy city.
- To solve this, we need to be able to turn this challenging complex problem into something a computer can handle
- But there's a vast gap between the two – the real world is filled with uncertainties and complex decisions
- While computer programs operates in a primarily deterministic way
- Bridging this chasm is one of the core challenges of AI

Paradigm

Modeling

Inference **Learning**

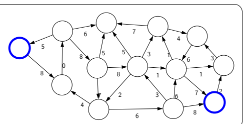
- In this course, we will adopt the **modeling-inference-learning** paradigm
- In this view, there are three pillars to AI: modeling, inference, and learning.

Paradigm: modeling

Real world

Modeling

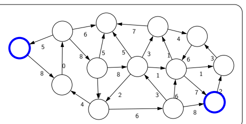
Model



- The first pillar is modeling.
- Modeling is the process of approximating real world problems using formal mathematical objects called **models**,
- By making problems precise, modeling allows us to study these problems and identify solutions using computers
- As an example, we might formulate the route finding problem as a graph where cities are vertices, edges represent the roads, and the cost of an edge represents the traffic on that road
- However, notice that this modeling process is lossy: not all of the richness of the real world can be captured.
- This is one key challenge in modeling – what complexity do we need to keep?
- There are some exceptions: games such as Chess, Go, or Sudoku are defined in a formal way so that the model are identical to the problem

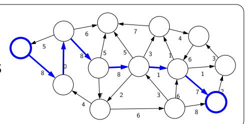
Paradigm: inference

Model



Inference

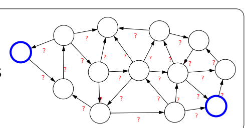
Predictions



- The second pillar is inference.
- Given a model, the task of **inference** is to answer questions about model.
- For example, given the model of the city, we might ask: what is the shortest path? or cheapest path?
- The focus of inference is usually on efficient algorithms that can answer these questions.
- For some models, computational complexity can be a concern (games such as Go), and usually approximations are needed.

Paradigm: learning

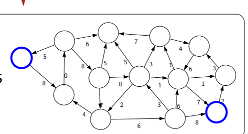
Model without parameters



+data

Learning

Model with parameters



- But where does the model come from?
- It seems daunting to write down a real, complex model by hand
- I for one would not be able to write down what traffic patterns look like in san francisco by hand
- This is where machine **learning** comes in
- Instead of constructing a full model,
- We write down a way to specify models abstractly
- For example a traffic model with unknown congestion parameters on each road
- If we can collect data about congestion, then we can **learn** what these unknown congestion paramters should be
- Machine learning is this process of turning an abstract model family that we can easily write down into a concrete model of the world that we can query

Course plan

"Low-level intelligence" "High-level intelligence"

Machine learning

- We now embark on our tour of the topics in this course.
- The topics correspond to types of models that we can use to represent real-world tasks.
- The topics will in a sense advance from low-level intelligence to high-level intelligence,
- We will go from models that make reflex decisison to models that are based on logical reasoning.

Machine learning

Data → Model

- The main driver of recent successes in AI
- Move from "code" to "data"
- Requires a leap of faith: **generalization**

- Supporting all of these models is **machine learning**,
- This has powered the recent success in AI.
- Why has machine learning had such an impact?
- Instead of writing ever more complex models, we can rely upon data to fill in the complexity
- This is way easier to obtain, especially with the internet
- The key observation to modern machine learning:
- When done properly, learning using data from the past gives us good models for the future
- Showign this formally is called statistically learning theory, and its a beautiful set of results we unfortunately wont get to cover in this course

Course plan

Reflex

"Low-level intelligence" "High-level intelligence"

Machine learning

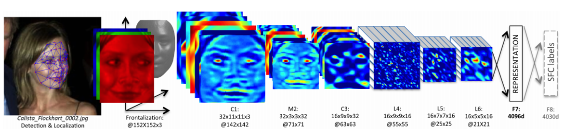
- We will now talk about some models – starting with ones that mimic human reflex
- To motivate this, im going to show you an example

What is this animal?

- What did you just see?
- Most of you could probably recognize the zebra in that split second.

Reflex-based models

- Examples: linear classifiers, deep neural networks



- Most common models in machine learning
- Fully feed-forward (no backtracking)

- A reflex-based model simply performs a fixed sequence of computations on a given input.
- Examples include most models found in machine learning, from simple linear classifiers to deep neural networks.
- The main characteristic of reflex-based models is that their computations are feed-forward so there's no backtracking and consideration of alternatives;
- Inferences like identifying animals is trivial because it is just running the fixed computations, which makes these models appealing.

Course plan

Search problems

Markov decision processes

Adversarial games

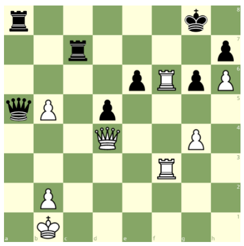
Reflex **States**

"Low-level intelligence" "High-level intelligence"

Machine learning

- Next, we will consider state-based models.

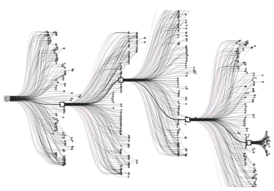
State-based models



White to move

- Consider the task of figuring out what move white should make given a particular chess position.
- Most of us will find this to require more thinking than just recognizing the zebra.

State-based models



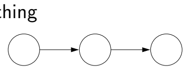
Applications:

- Games: Chess, Go, Pac-Man, Starcraft, etc.
- Robotics: motion planning
- Natural language generation: machine translation, image captioning

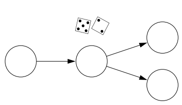
- Reflex-based models are too simple for tasks that require more planning, like chess
- State-based models overcome this limitation.
- The key idea is to think about states (like the configuration of the chessboard)
- as well as actions (what moves we can make using the pieces)
- Representing this as a graph gives us powerful tools to plan ahead:
- We can efficiently answer questions like - what move should I make to eventually get to a checkmate?

State-based models

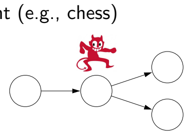
Search problems: you control everything



Markov decision processes: against nature (e.g., Blackjack)

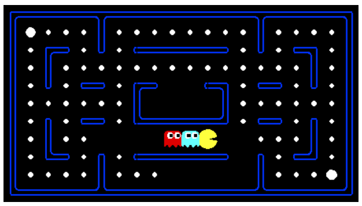


Adversarial games: against opponent (e.g., chess)



- **Search problems** are adequate models when you are operating in an environment that has no uncertainty. However, in many realistic settings, there are other forces at play.
- **Markov decision processes** are ways of expanding the earlier approach to handle tasks with an element of chance (e.g., Blackjack).
- **Adversarial games**, as the name suggests, handle tasks where there is an opponent who is actively working against you (e.g., chess).

Pac-Man



[demo]

- In one of the homeworks, you will build an agent that can play Pac-Man.
- To whet your appetite, this is what it will look like.
- (demo)
- Think back to the state-based approach: what should the states of the model be?
- What should the transitions that we can take look like?
- Is there an adversary? is there randomness?

Course plan

Search problems — Constraint satisfaction problems

Markov decision processes — Markov networks

Adversarial games — Bayesian networks

Reflex **States** **Variables**

"Low-level intelligence" "High-level intelligence"

Machine learning

- Next, we will talk about variable-based models.

Sudoku

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

→

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

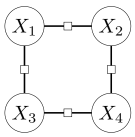
Goal: put digits in blank squares so each row, column, and 3x3 sub-block has digits 1–9

Key: order of filling squares doesn't matter in the evaluation criteria!

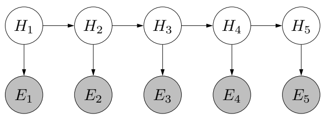
- In state-based models, solutions are actions: they specify step by step instructions that must be followed in order
- In some applications, the order isn't important.
- For example, consider Sodoku. The goal of this pouzzle is to put digits in the blank squares to satisfy uniqueness constraints,
- All that matters is the final configuration of numbers; so you can fill them in in any order.
- Casting this as a search problem is wildly ineffiicient. We should leverage the fact that order doesnt matter

Variable-based models

Constraint satisfaction problems: hard constraints (e.g., Sudoku, scheduling)



Bayesian networks: soft dependencies (e.g., tracking cars from sensors)



- **Constraint satisfaction problems** are variable-based models where we only have hard constraints. For example, in scheduling, one person can't be in two places at once.
- **Bayesian networks** are variable-based models where we want to find **plausible** assignments of variables
- For example, if we're tracking a car using video, the position is likely to not change too much frame to frame
- This isn't a hard constraint – we can strap a jet engine to a car to violate it – but it's a constraint that holds most of the time
- This kind of structure if often reprsented as a dependency graph between random variables.

Course plan

Search problems | Constraint satisfaction problems
Markov decision processes | Markov networks
Adversarial games | Bayesian networks

Reflex **States** **Variables** **Logic**

"Low-level intelligence" "High-level intelligence"

Machine learning

- The last topic is logic.

Motivation: virtual assistant

Tell information

Ask questions

Use natural language!

[demo]

Need to:

- Digest **heterogenous** information
- Reason **deeply** with that information

- Logic is one of the foundational topics of AI, but has unfortunately often been passed over as being not useful
- I want to end this survey discussing a concrete setting where we want more logically consistent behaviors
- Consider a virtual assistant. A good one should remember various facts youve told it and answer questions that require drawing inferences from its knowledge.
- I'll show you a demo which you'll have an opportunity to play with in the final homework.
- (demo)
- Interacting with this system feels very different than a typical machine learning-based system in a few ways.
- First, it is adaptive, whereas most ML systems are a fixed function.
- Second, our queries are heterogenous and more abstract, requiring logical reasoning to get the right answer
- One often contrasts logical AI and statistical AI, but they are complementary tools.
- statistical AI is popular and successful now, but remember that whats popular now is unlikely to stay so forever
- Neural nets, which are dominant now was not a mainstream topic until the 2010s

Course plan

Search problems
Markov decision processes
Adversarial games

Constraint satisfaction problems
Markov networks
Bayesian networks

Reflex **States** **Variables** **Logic**

"Low-level intelligence" "High-level intelligence"

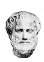
Machine learning

- And this concludes our tour of the topics.
- AI includes a broad range of models and complementary approaches
- This includes reflexive ones such as image recognition that perform well in noisy real-world settings, as well as first-order-logic based ones that handle hard constraint-based problems

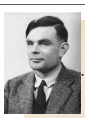
What's next?

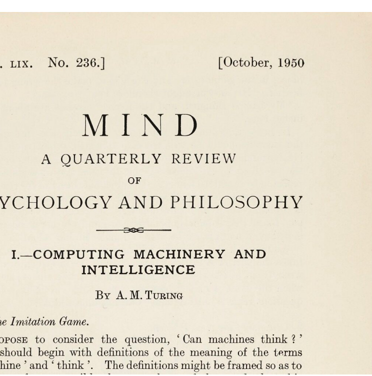
 Course logistics

 Course content

 AI history

 AI today





. LIX. No. 236.] [October, 1950

MIND

A QUARTERLY REVIEW

OF

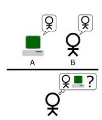
PSYCHOLOGY AND PHILOSOPHY

I.—COMPUTING MACHINERY AND INTELLIGENCE

By A. M. Turing

1. *The Imitation Game.*

I propose to consider the question, 'Can machines think?' This should begin with definitions of the meaning of the terms 'machine' and 'think'. The definitions might be framed so as to



objective specification

Many people think that a very abstract activity, like the playing of chess, would be best. It can also be maintained that it is best to provide the machine with the best ***sense organs*** *that money can buy, and then teach it to understand and speak English. This process could follow the normal* ***teaching of a child****. Things would be pointed out and named, etc. Again I do not know what the right answer is, but I think both approaches should be tried.*

- A natural place to start talking about the history of AI is Alan Turing's landmark 1950 paper called Computing Machines and Intelligence.
- In this paper, Turing asked the question, "Can machines think?" and answered it with the Imitation Game, or the Turing Test.
- There are many versions and interpretations – the original version from Turing is not the standard one that is often discussed
- Ill talk about the standard version – which is that a interrogator passes a series of written questions to a human and a machine, and the goal of the interrogator is to identify which is human and which is machine
- This paper is remarkable not because it built a system or proposed any methods, but because it operationalized intelligence as something that could be quantified:
- The distinguishability of human and machine answers serves as a way to measure the complex and ill-defined idea of human intelligence
- For us, one important takeaway of the Turing test is that the **goal** of intelligence can be decoupled from the methods that might get us there. This means that there are many approaches that might lead to success
- At the end of the paper, Turing discusses two possible approaches. The first is based on solving abstract problems like chess, which is the route taken by symbolic AI. The second is where you build a machine and teach like a child, which is the route taken by neural and statistical AI.
- I will now tell three stories of symbolic, neural, and statistical AI.

1956

- 1956 is the beginninig of our first story.

Birth of AI

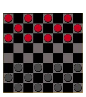
1956: John McCarthy organized workshop at Dartmouth College

Every aspect of learning or any other feature of intelligence can be so precisely described that a machine can be made to simulate it.

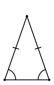
general principles

- It is the year that the name **artifical intelligence** was coined.
- John McCarthy, who later founded the Stanford AI lab, organized a workshop at Dartmouth College that summer.
- In addition to McCarthy, the workshop was attended by Marvin Minsky, Allen Newell, Herbert Simon, etc., all of whom went on to make seminal contributions in AI.
- The participants laid out a bold proposal: to build a system that could capture every aspect of intelligence.
- This is what we refer today as **general** intelligence.
- Indeed, during this post-war era, computers were just coming on the scene. It was a very exciting time and people were ambitious.
- The proposal for the workshop suggested fundamental advances could be made in 2 months

Birth of AI, early successes



Checkers (1952): Samuel's program learned weights and played at strong amateur level



Problem solving (1955): Newell & Simon's Logic Theorist: prove theorems in Principia Mathematica using search + heuristics; later, General Problem Solver (GPS)

- There were early, dramatic successes to back this view up
- Arthur Samuel wrote a program that could play checkers at a strong amateur level.
- Alan Newell and Herbert Simon's Logic Theorist could prove theorems. For one theorem, it actually found a proof that was more elegant than the human-written proof.
- They tried to publish a paper on the result, but unfortunately the paper got rejected because it was not a new theorem.
- Later, they developed the General Problem Solver, which promised to solve any problem (which could be suitably encoded in logic), again carrying forward the ambitious "general intelligence" agenda.
- Progress on hard games like checkers and reasoning tasks like proofs seemed promising

Overwhelming optimism...

Machines will be capable, within twenty years, of doing any work a man can do.

Within 10 years the problems of artificial intelligence will be substantially solved.

I visualize a time when we will be to robots what dogs are to humans, and I'm rooting for the machines.

- With these initial success, it was a time of high optimism, with all the leaders of the field, all impressive thinkers, predicting that AI would be "solved" in a matter of years.

...underwhelming results

Example: machine translation

The spirit is willing but the flesh is weak.

↓

(Russian)

↓

The vodka is good but the meat is rotten.

1966: ALPAC report cut off government funding for MT, first AI winter

- Despite successes in domains like games, more real-world tasks like machine translation were complete failures.
- There is a folklore story of how the sentence "The spirit is willing but the flesh is weak" was translated into Russian and then back to English, leading to the amusing translation "The vodka is good but the meat is rotten".
- However, this translation was not so amusing to government agencies funding the research.
- And so in 1966, the ALPAC report resulted in funding being cut off for machine translation.
- This marked the beginning of the first AI winter.

Implications of early era

Problems:

- **Limited computation**: search space grew exponentially, outpacing hardware
- **Limited information**: complexity of AI problems (number of words, objects, concepts in the world)

Useful contributions (John McCarthy):

- Lisp
- Garbage collection
- Time-sharing

- What went wrong? Two things.
- The first was computation.
- Most of the approaches casted problems as logical reasoning, which required a search over an exponentially large search space. Combined with hardware limitations, this meant complex problems were out of reach
- The second is information and modeling. AI systems at the time required manually entering facts and models of the world The real world is simply too complex, with innumerable facts and events to do this by hand
- Though the grand ambitions were not realized, some generally useful technologies came out of the effort.
- Lisp was way ahead of its time in terms of having advanced language features.
- People programming in high-level languages like Python take garbage collection for granted.
- And the idea that a single computer could simultaneously be used by multiple people (time sharing) was prescient.

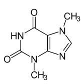
Knowledge-based systems (70-80s)

Expert systems: elicit specific domain knowledge from experts in form of rules:

```
if [premises] then [conclusion]
```


- In the 1970s and 80s, AI researchers looked to knowledge as a way to combat both the computation and information limitations of the previous era.
- At this time, expert systems became fashionable, where a domain expert would encode their domain expertise in these systems, usually in the form of if-then rules.

Knowledge-based systems (70-80s)

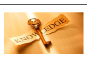


DENDRAL: infer molecular structure from mass spectrometry

MYCIN: diagnose blood infections, recommend antibiotics

XCON: convert customer orders into parts specification

- There was also a noticeable shift in focus.
- Instead of the solve-it-all optimism from the 1950s and 60s, researchers focused on building narrow practical systems in targeted domains.
- By focusing on narrower domains, and fully encoding knowledge within the domains these systems were limited but useful
- Famous examples from this era included systems for chemistry, medical diagnosis, and business operations.



Knowledge-based systems

Wins:

- Knowledge helped both the **information** and **computation** gap
- First **real application** that impacted industry

Problems:

- Deterministic rules couldn't handle the **uncertainty** of the real world
- Rules quickly became too **complex** to create and maintain

A number of people have suggested to me that large programs like the SHRDLU program for understanding natural language represent a kind of dead end in AI programming. Complex interactions between its components give the program much of its power, but at the same time they present a formidable obstacle to understanding and extending it. In order to grasp any part, it is necessary to understand how it fits with other parts, presents a dense mass, with no easy footholds. Even having written the program, I find it near the limit of what I can keep in mind at once. — Terry Winograd

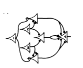
1987: Collapse of Lisp machines and second AI winter

- What knowledge (in addition to the restriction to narrow domains) did was not only providing information to the system.
- Prior approaches considered every possiblity using logical inference, but these new systems used knowledge to search only over likely solutions.
- Also, this was the first time AI had a real impact on industry, rather than being just an academic's playground.
- However, knowledge engineering ran into major limitations. First, deterministic rules failed to capture the uncertainty in the real world,
- Second, these systems were just too much work to create and maintain, making it hard to scale up to more complex problems.
- Terry Winograd built a famous dialogue system called SHRDLU summed up well by the sentiment in this quote: the complex interactions between all the components made it too hard for mortals to even grasp. After that, he moved to Stanford and became an HCI professor.
- During the 80s, there was again a lot of overpromising and underdelivering, the field collapsed again. It seemed like history was repeating itself.
- We will now leave the story of symbolic AI, which dominated AI for multiple decades...

1943

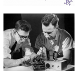
- ...and go back in time to to 1943 to tell the story of neural AI.

Artificial neural networks

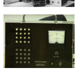


1943: artificial neural networks, relate neural circuitry and mathematical logic (McCulloch/Pitts)

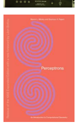
1949: "cells that fire together wire together" learning rule (Hebb)



1958: Perceptron algorithm for linear classifiers (Rosenblatt)



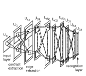
1959: ADALINE device for linear regression (Widrow/Hoff)



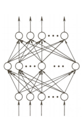
1969: Perceptrons book showed that linear models could not solve XOR, killed neural nets research (Minsky/Papert)

- In 1943, neurophysiologist Warren McCulloch and logician Walter Pitts devised a simple mathematical model of a neuron, giving birth to the field of (artificial) neural networks.
- They showed how this model could compute arbitrary logical functions (and, or, not, etc.), but did not suggest a method for learning this model.
- In 1949, neuropsychologist Donald Hebb introduced the first learning rule. It was based on the intuition that cells that fire together wire together. This rule was nice in that it was local, but it was unstable and so didn't really work.
- In 1958, Frank Rosenblatt developed the Perceptron algorithm for learning single-layer networks (a.k.a. linear classifiers), and built a device that could recognize simple images.
- In 1959, Bernard Widrow and Ted Hoff came up with ADALINE, a different learning rule corresponding to linear regression. A multi-layer generalization called MADALINE was used later to eliminate echo on phone lines, one of the first real-world applications of neural networks.
- 1969 was an important year. Marvin Minsky and Seymour Papert published a book that explored various mathematical properties of Perceptrons. One of the (trivial) results was that the single-layer version could not represent the XOR function. Even though this says nothing about the capabilities of deeper networks, the book is largely credited with the demise of neural networks research, and the continued rise of symbolic AI.

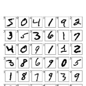
Revival of connectionism



1980: Neocognitron, a.k.a. convolutional neural networks for images (Fukushima)



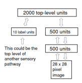
1986: popularization of backpropagation for training multi-layer networks (Rumelhardt, Hinton, Williams)



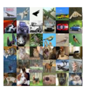
1989: applied convolutional neural networks to recognizing handwritten digits for USPS (LeCun)

- In the 1980s, there was a renewed interest in neural networks under the banner of connectionism, and there were many new links to psychology and cognitive science.
- The Neocognitron developed by Kunihiko Fukushima was the first convolutional neural network, with multiple layers and pooling. It was trained in a rather heuristic way.
- Donald Rumelhardt, Geoff Hinton, and Ronald Williams rediscovered (yet again) and popularized backpropagation as a way to train multi-layer neural networks, and showed that the hidden units could capture interesting representations.
- Yann LeCun built a system based on convolutional neural networks to recognize handwritten digits. This was deployed by the USPS to recognize zip codes, marking one of the first success stories of neural networks.

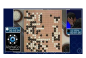
Deep learning



2006: unsupervised layerwise pre-training of deep networks (Hinton et al.)



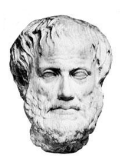
2012: AlexNet obtains huge gains in object recognition; transformed computer vision community overnight



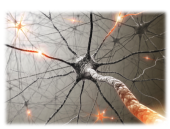
2016: AlphaGo uses deep reinforcement learning, defeat world champion Lee Sedol in Go

- But until the mid-2000s, neural network research was still quite niche, and they were still notoriously hard to train.
- The perception was that they achieved good performance, but was a black box and took enormous engineering effort to get working
- In 2006, this started changing when Geoff Hinton and colleagues published a paper showing that it was possible to train neural nets with many many layers, which they referred to as deep nets The term deep learning started around this time.
- The real break for neural networks came in the 2010s. In 2012, Alex Krizhvesky, Ilya Sustkever, and Geoff Hinton trained a landmark convolutional neural network called AlexNet, which resulted in massive improvements on the ImageNet benchmark, the gains were so substantial that it convinced a large part of hte vision community
- In 2016, DeepMind's AlphaGo was another turning point. By defeating humans at Go, a feat that many experts thought was still a few decades away, deep learning firmly established itself as the dominant paradigm in AI.

Two intellectual traditions



symbolic AI



neural AI

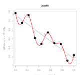
Food for thought: deep philosphical differences, but deeper connections (McCulloch/Pitts, AlphaGo)?

- So far, we've seen two intellectual traditions, symbolic AI, with roots in logic and neural AI, with roots in neuroscience.
- While the two have fought fiercely over deep philosophical differences, they share similar goals and perhaps there are deeper connections.
- For example, McCulloch and Pitts' work from 1943 can be viewed as the root of deep learning, but that paper is mostly about how to implement logical operations.
- The game of Go can be perfectly characterized by a set of simple logic rules. But AlphaGo did not tackle the problem directly using logic and instead leveraged the pattern matching capabilities of artificial neural networks.

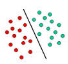
1801

- But there's a third and final story we must tell to complete the picture. This story is not really about AI per se, but rather the influx of certain other areas that have helped build a solid mathematical foundation for AI. This **statistical AI** perspective is also how we will frame the topics in this course.

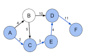
Early ideas from outside AI



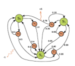
1801: linear regression (Gauss, Legendre)



1936: linear classification (Fisher)



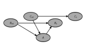
1956: Uniform cost search for shortest paths (Dijkstra)



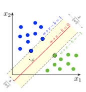
1957: Markov decision processes (Bellman)

- The idea of fitting models from data, which is at the heart of machine learning and modern AI, goes back to as far as Gauss and Legendre, who developed the principle of least squares for linear regression.
- Classification (linear discriminant analysis) was developed by Fisher in statistics.
- In general, machine learning has quite a bit of overlap with the statistics and data mining communities, who worked on solving concrete problems without the lofty goals of "intelligence".
- Outside of statistics and machine learning, AI consists of sequential decision making problems. Along these lines, there's Dijkstra's algorithm for finding shortest paths for deterministic settings.
- Bellman developed Markov decision processes in the context of control theory, which handles uncertainty in the world.
- Note that these developments largely predated AI.

Statistical machine learning



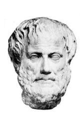
1985: Bayesian networks (Pearl)



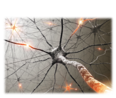
1995: Support vector machines (Cortes/Vapnik)

- You might have noticed that our story of symbolic AI ended at the end of the 1980s, but neural AI only became widespread in the 2010s.
- What happened in the intervening 30 years?
- This is because for much of the 1990s and 2000s, the term AI wasn't actually used as much as it is today, partly as a way to distance the field from the overpromises made in early AI
- People talked about **machine learning** instead, and during that time period, machine learning was dominated by two paradigms.
- The first is Bayesian networks, developed by Judea Pearl, which provides an elegant framework for **reasoning under uncertainty**, something that symbolic AI didn't have a satisfying answer for.
- The second is Support Vector Machines (SVMs), which originated from statistical learning theory and optimization. Unlike neural nets, it was possible to get provable guarantees for optimizing and learning SVMs and became the favored tool in machine learning.

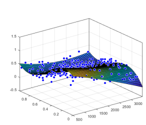
Three intellectual traditions



symbolic AI



neural AI



statistical AI

- This concludes our tour of the three stories that make up what AI is today.
- **Symbolic AI** took a top-down approach and failed to fulfill its original promise. But it offered a vision and did built impressive artifacts for ambitious problems like question answering and dialogue systems along the way.
- **Neural AI** took a completely different approach, proceeding bottom-up, starting with simple perceptual tasks, which the symbolic AI community wasn't interested in. It offered a class of models, deep neural networks, that with today's data and computing resources, has proven capable of conquering ambitious problems.
- Finally, **statistical AI** foremost offers mathematical rigor and clarity. It provides methods for analyzing and understanding the behavior of AI systems trained using real-world data Even when we are not using statistical methods, the language of statistical AI often allows us to precisely state the goals and assumptions of our models Because of this, this course will be largely presented through the lens of statistical AI.
- Stepping back, the modern world of AI is like New York City—it is a melting pot that has drawn from many different fields ranging from statistics, algorithms, neuroscience, optimization, economics, etc. And it is the symbiosis between these fields and their application to important real-world problems that makes working in AI so rewarding.

Further reading

Wikipedia article: https://en.wikipedia.org/wiki/History_of_artificial_intelligence

Encyclopedia of Philosophy article: https://plato.stanford.edu/entries/artificial-intelligence

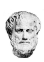
Turing's Computing Machinery and Intelligence: https://www.csee.umbc.edu/courses/471/papers/turing.pdf

History and Philosophy of Neural Networks: https://research.gold.ac.uk/10846/1/Bishop-2014.pdf

What's next?

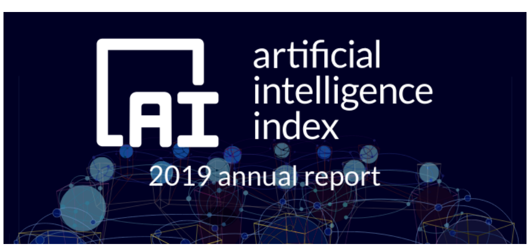
 Course logistics

 Course content

 AI history

 AI today

- If there were one word to describe the state of AI today, it would be "surreal."
- It's hard to imagine that just ten years ago, AI was a term people actively avoided using because of its connections to AI winter And now there are national AI strategies being formed



Prior to 2012, AI results closely tracked Moore's Law, with compute doubling every two years. Post-2012, compute has been **doubling every 3.4 months**.

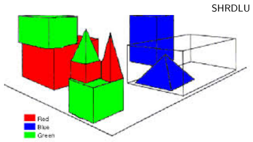
In 2019, the largest AI conference, **NeurIPS**, *expects 13,500 attendees, up 41% over 2018 and over* **800% relative to 2012**.

In the US, the share of **jobs** *in AI-related topics increased from* **0.26% of total jobs posted in 2010 to 1.32% in October 2019**.

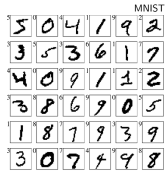
- The AI index is a project hosted out of Stanford which aims to track the state of AI in a data-driven way.
- Each year, it releases a report summarizing not just research progress, but also conference participation, impact on the economy, education, etc.
- Here are some quotes from the report: compute (mostly GPUs) has been doubling every 3.4 months.
- NeurIPS, the flagship machine learning conference has grown by 8x over the last 8 years.
- It used to fit in a single hotel ballroom, now it's an entire two floors of a convention center and tickets sell out in 12 minutes
- And the number of AI jobs has increased by 5x.

In vitro

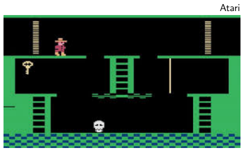
SHRDLU



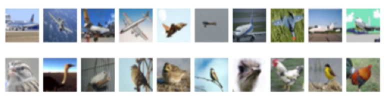
MNIST



Atari



CIFAR-10



- One of the biggest changes we've witnessed in AI is the transition from the lab to the real-world.
- For a long time, AI was limited to relatively artificial environments and datasets, which was (and is) still useful to spur the development of new methods.
- But now we are seeing much more real-world deployment of AI in ways that have a direct impact on people's lives.
- It's important to note that AI, like any technology, is an amplifier. It makes what is good better, and it makes what is bad worse. We need to be aware of both sides.

Prospects

- Let us start with the positive side.

Virtual assistants

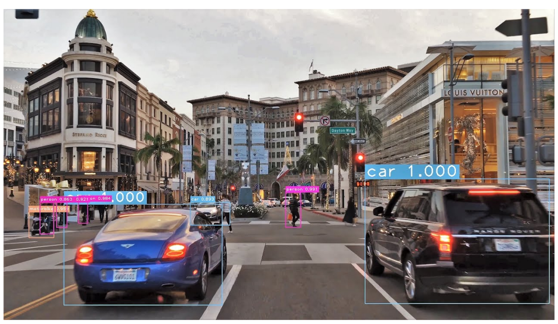
- In the last decade, speech recognition and question answering have improved remarkably.
- You can now talk to your favorite virtual assistant and expect some basic (though obviously not perfect) level of language understanding. And children are growing up thinking that talking to computers is normal.
- Search engines such as Google have already demonstrated the enabling power that comes with having the world's information at one's fingertips.
- Providing this through a natural language interface makes it more efficient and natural, and could have the potential to be especially be useful for people who might not have the means to use a computer (if they were designed for that population).

Machine translation

Input sentence:	Translation (PBMT):	Translation (GNMT):	Translation (human):
李克強此行將啟動中加總理年度對話機制，與加拿大總理杜魯多舉行兩國總理首次年度對話。	Li Keqiang premier added this line to start the annual dialogue mechanism with the Canadian Prime Minister Trudeau two prime ministers held its first annual session.	Li Keqiang will start the annual dialogue mechanism with Prime Minister Trudeau of Canada and hold the first annual dialogue between the two premiers.	Li Keqiang will initiate the annual dialogue mechanism between premiers of China and Canada during this visit, and hold the first annual dialogue with Premier Trudeau of Canada.

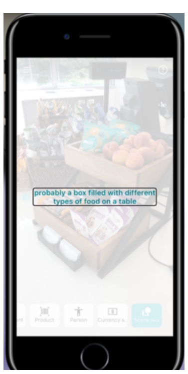
- Language barriers pose significant challenges to immigrants, travellers, businesses, and minority subcommunities, both in terms of connecting with others but also access to valuable information.
- Machine translation aims to overcome these barriers.
- Machine translation has made huge strides since the 1960s, and while it is not perfect, it is good enough for someone to get the basic gist of a document written in another language and to even communicate with another person speaking another language in real-time.

Autonomous driving



- Autonomous vehicles have the potential to one day significantly reduce accidents and congestion on our roads.
- A prerequisite is that the car doesn't hit anything, and one of the ways that it figures out what's in front is using a camera.
- Computer vision has made swift progress towards recognizing and localizing objects in relatively unstructured scenes, but there is still some headroom to achieve the needed reliability.

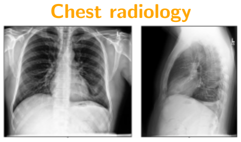
Visual assistive technology



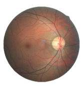
- This example is the Seeing AI app from Microsoft Research, which narrates whatever the camera is pointed at.
- This visual assistive technology could be a game-changer for the visually impaired.
- Conversely, auto-captioning technology, which turns sound into sight, is potentially also quite useful for the hearing-impaired.

Healthcare

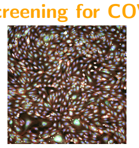
Chest radiology



Diabetic retinopathy



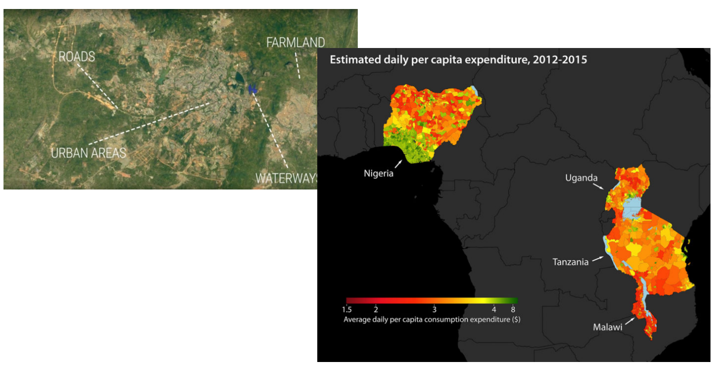
Drug screening for COVID-19



- AI for healthcare is also an area of growing importance, both for diagnosis and for therapeutic development, especially in areas in the world with a shortage of clinical specialists.
- One example is interpreting chest x-rays for detecting diseases such as pneumonia and collapsed lung.
- Another is diagnosing diabetic retinopathy, which causes blindness in diabetic patients.
- Finally, there's a recent dataset with experiments showing how COVID-19 infected cells respond to certain drugs, with the hope that one can find drugs that can treat late-stage COVID-19.

[Jean et al. 2016]

Poverty mapping

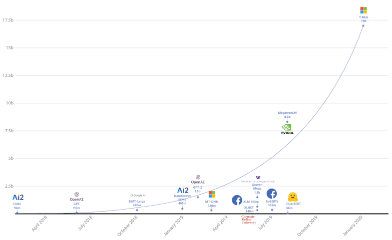


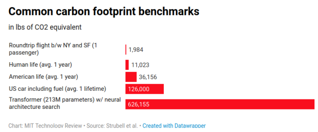
- At a more societal level, it is well-known that poverty is a huge problem in the world, with more than 700 million people living in extreme poverty according to the World Bank.
- But even identifying the areas in greatest need is challenging due to the difficulty of obtaining reliable survey data.
- Some work has shown that satellite images (which are readily available) can be used to predict various wealth indicators based on the types of roofs or presence of roads or night lights.
- This information could be informative for governments and NGOs to take proper action and monitor progress.

Risks

- This all sounds great, so what's the catch?

Energy consumption





GPT-3 (released May 2020) from OpenAI has **175 billion** parameters

- First, there is the cost of training high-performing models.
- If we look at NLP, people realized that making models bigger resulted in better performance, this has led to an arms-race of model scaling to somewhat absurd levels.
- In April 2018, models had about 100 million parameters.
- BERT, which made a big splash, had 340 million parameters.
- In January 2020, Microsoft released a 17 billion parameter model.
- In May 2020, OpenAI released GPT-3, which was ten times larger still, which, if placed on the chart, would be way off the screen.
- There was a paper that examined the environmental impact of training deep learning models. A 213 million parameter model with neural architecture search was about 5 times the CO2 emissions during the lifetime of a US car. I'll let you speculate about the energy consumption of GPT-3.
- The question this raises is – if these systems become ubiquitious, will their demand for energy lead to environmental harms?

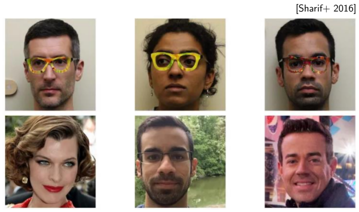
Privacy

- Machine learning algorithms rely heavily on readily accessible, large datasets.
- More data leads to better performance, so companies and nation-states have strong incentives to collect as much data as they can
- Coupled with our mobile devices which can generate a wealth of information, this leads to a dangerous situation where the desire to train ML systems incentivizes violating people's right to privacy
- This has led to emerging work in privacy-preserving machine learning, which allows data and learning to happen on devices in a decentralized way, and only transmit limited statistics to a central server.

Security

[Evtimov+ 2017]

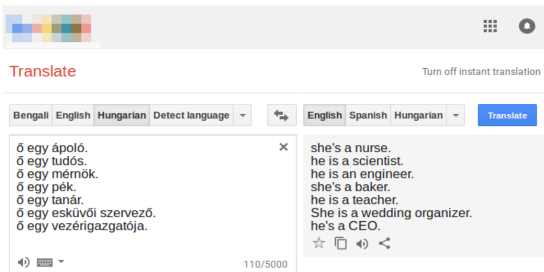
[Sharif+ 2016]



- In high-stakes applications such as autonomous driving and authentication (face ID), models need to not only be accurate but need to be robust against **attackers**.
- Researchers have shown how to generate **adversarial examples** to fool systems.
- For example, you can put stickers on a stop sign to trick a computer vision system into mis-classifying it as a speed limit sign.
- You can also purchase special glasses that fool a system into thinking that you're a celebrity.
- Guarding against these attackers is a wide open problem.

[Prates+ 2018]

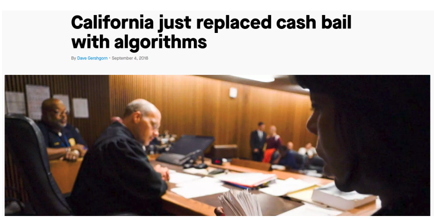
Bias



- A less spectacular but maybe more pernicious problem is bias.
- In this example, if you take Hungarian, in which "he" and "she" are not differentiated, and translate into English, the model has to hallucinate gender from context.
- This experiment reveals the various gender stereotypes that the model harbors.
- Even though machine learning algorithms are based on mathematical principles, the usual guarantees we get from learning theory is about **accuracy**
- This means there are a range of harms that can still happens - models that reflect biases in the data, or models that have errors that are more harmful to minorities

Fairness

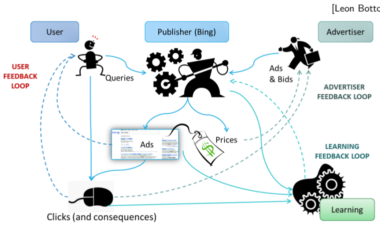
- Northpointe: COMPAS predicts criminal risk score (1-10)
- ProPublica: given that an individual did not reoffend, Black people 2x likely to be (wrongly) classified 5 or above
- Northpointe: given a risk score of 7, 60% of White people reoffended, 60% of Black people reoffended

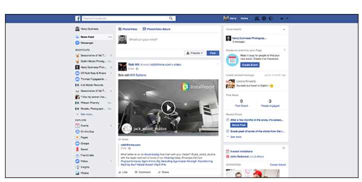


- The stakes are increased when machine learning models are used to make decisions that impact someone's life in a serious way (e.g., bail, loans, hiring).
- There was one instance where Northpointe created a software system called COMPAS to predict recidivism risk, i.e., whether they will commit a crime again.
- ProPublica, a nonprofit organization that does investigative journalism, found that given that an individual didn't commit a crime, Black people were were predicted to have higher risk, and therefore, unfair.
- Northpointe rebutted, noting that given a risk score of 7, the probability of commiting a crime was the same for both Black and White people, and therefore it was fair.
- These just happen to be two different notions of fairness (equalized odds versus predictive rate parity). Unfortunately, there is no "right answer," and it has even been shown mathematically that it is impossible for a imperfect classifier to satisfy three notions of fairness simultaneously (Kleinberg et al., 2016).
- This indicates that making progress on fairness is not just a technical challenge, but one that requires social and policy considerations.

Feedback loops in learning

[Leon Bottou]





- Relatedly, AI systems are deployed in a dynamic environment in which the system makes predictions (search results, recommendations, ads), users take action (e.g., clicks).
- These actions are recorded as data used to retrain the system, often to reinforce these actions.
- This introduces a feedback which usually amplifies or polarizes the initial signal, leading to unstable behavior.
- An open research challenge is to design learning algorithms whose dynamics are stable.

Generating fake content

In order to get something done, maybe we need to think less. Seems counter-intuitive, but I believe sometimes our thoughts can get in the way of the creative process. We can work better at times when we "tune out" the external world and focus on what's in front of us.

I've been thinking about this lately, so I thought it would be good to write an article about it.

So what exactly does this mean? Well, for starters, let's start with some definitions.

Definition #1: Creative Thinking (CT) is the act of using your mind to come up with ideas that are not already present in reality or that have never existed before. It is a form of mental activity that involves the use of logic and reason.

Definition #2: Over-Thinking (OT) is the act of trying to come up with ideas that have already been thought through by someone else. OT usually results in ideas that are impractical, impossible, or even stupid.

- Finally, Deep learning has enabled us to generate surprisingly realistic content.
- On the left, is an Obama deepfake, a video that shows Obama saying things he never did.
- On the right is part of a blog post that a student used GPT-3 to generate, which ended up ranked number one on Hacker News for a while.
- It is clear that we have already lost the ability to tell the difference between real and fake content. Furthermore, given the ease and scale that fake content can be generated, bad actors spreading disinformation is a big threat to our society.

Prospects and risks of AI

AI technology is an amplifier

- Can reduce accessibility barriers and improve efficiency
- Can amplify bias, security risks, centralize power

Can build it $\neq$ should build it

- To conclude, it is worth stressing that AI, or technology in general, is an amplifier.
- We've seen that it can reduce accessibility barriers and improve the lives of the less fortunate.
- On the other hand, there are biases, new security threats, and a danger of centralization of power due to AI as well. We've only skimmed the surface here; there is much more to be said on this topic.
- As you proceed through this course, I would urge you to keep these issues in mind. If we are not careful, we could end up doing more harm than good. Just because you can build it doesn't mean you should.
- Figuring out the right way to reap the benefits and mitigate the risks will also require having a deep technical understanding, especially to develop novel solutions, which is what this course seeks to provide.