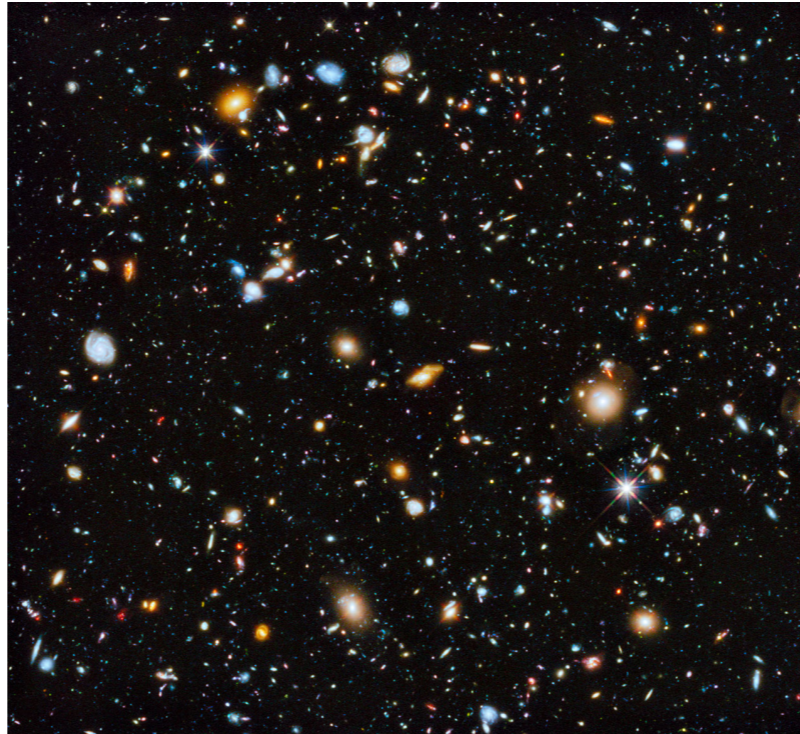




Lecture 1: Introduction





Roadmap

AI history

Ethics and responsibility

Course content

- I will present a short history of AI, which will necessarily be simplified and incomplete. But, I hope that it will give you a general appreciation of AI's multi-faceted history.



. LIX. No. 236.]

[October, 1950

MIND

A QUARTERLY REVIEW

OF

PSYCHOLOGY AND PHILOSOPHY

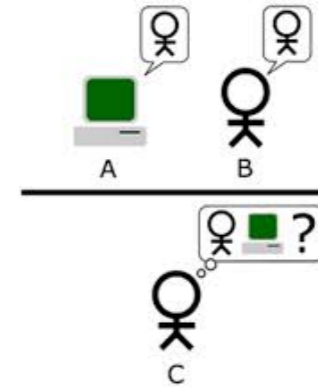


I.—COMPUTING MACHINERY AND INTELLIGENCE

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 paper published in 1950 called Computing Machines and Intelligence.
- In this paper, Turing asked the question, "Can machines think?" and answered it with the Imitation Game, more commonly known as the Turing Test. As some of you might know, a machine is said to pass the Turing test if it can convince a human judge that it's actually a human through natural language dialogue.
- This paper is remarkable not because it built a system or proposed any methods, but because it framed the philosophical discussions for decades to come. You have to appreciate how difficult a notion like intelligence is to pin down. So this was really the first actionable, formal answer to the question, "Can machines think?"
- Whether passing the Turing test is something that should be directly worked on is questionable and controversial, but the philosophical implications are quite thought-provoking.
- For us, one important takeaway of the Turing test is the separation of the **objective** specification of what we want a system to do (the "what") from the methods that might get us there (the "how"). This decoupling is a major theme throughout this course.
- 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 beginning of our first story.

Birth of AI

1956: John McCarthy organized workshop at Dartmouth College

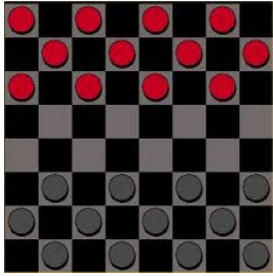


Every aspect of learning or any other feature of intelligence can be so precisely described that a machine can be made to simulate it.

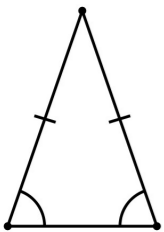
general principles

- It is the year that the name **artificial 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. They were after **generality**.
- Indeed, during this post-war era, computers were just coming on the scene. It was a very exciting time and people were ambitious.

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)

- A few notable systems were created during this time.
- 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 the paper got rejected because it was not a new theorem. Perhaps the reviewers failed to realize that the third author was actually a computer program.
- 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.

Overwhelming optimism...

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

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

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

- 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 the successes, certain tasks such as 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. 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 cast problems as logical reasoning, which required a search over an exponentially large search space. The hardware at the time was simply too limited.
- The second is information. Even if researchers had infinite computation, AI would not have been solved. There are simply too many concepts, words, and objects in the world, and this information has to somehow be put into the AI system.
- 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)

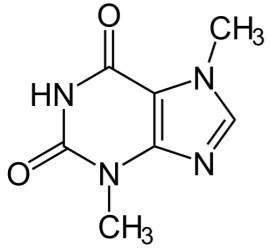


Expert systems: elicit specific domain knowledge from experts in form of rules:

```
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.
- 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*

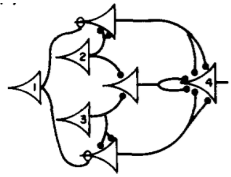
1987: Collapse of Lisp machines and second AI winter

- What knowledge (in addition to the restriction to narrow domains) did was not only providing information to the system, but it also helped alleviate the need for as much computation, by placing constraints on the space of possibilities.
- 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, though there were attempts to patch this heuristically as an afterthought.
- 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...

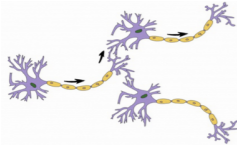
1943

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

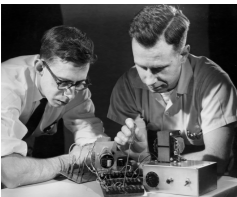
Artificial neural networks



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



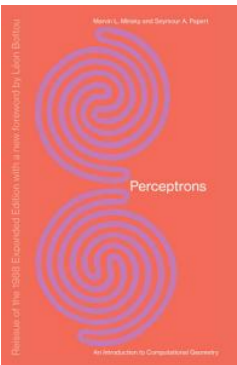
1949: "cells that fire together wire together" learning rule (Hebb)



1958: Perceptron algorithm for linear classifiers (Rosenblatt)



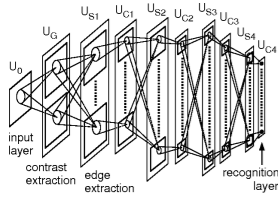
1959: ADALINE device for linear regression (Widrow/Hoff)



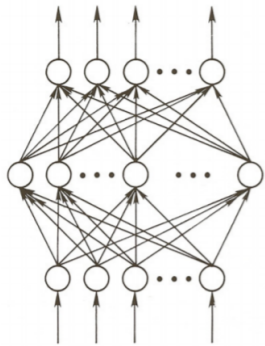
1969: Perceptrons book showed that linear models could not solve XOR, killed neural nets research (Minsky/Papert)

- 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.

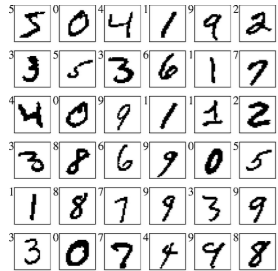
Revival of connectionism



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



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

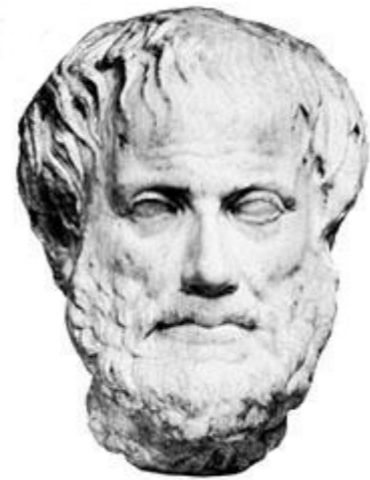


1989: applied convolutional neural networks to recognizing handwritten digits for USPS (LeCun)

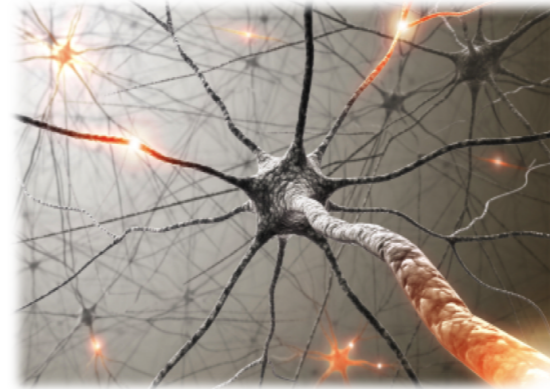
- In the 1980s, there was a renewed interest in neural networks under the banner of connectionism, and there were many new links to psychology and cognitive science.
- The Neocognitron developed by Kunihiko Fukushima was the first convolutional neural network, with multiple layers and pooling. It was 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.

- But until the mid-2000s, neural network research was still quite niche, and they were still notoriously hard to train. In 2006, this started changing when Geoff Hinton and colleagues published a paper showing how deep networks could be trained in an unsupervised manner, and then fine-tuned on a small amount of labeled data. The term deep learning started around this time. This "pre-training" technique is ubiquitous today.
- The real break for neural networks came in the 2010s. In 2012, Alex Krizhvesky, Ilya Sutskever, and Geoff Hinton trained a landmark convolutional neural network called AlexNet, which resulted in massive improvements on the ImageNet benchmark, turning the skeptical computer vision community into believers almost instantaneously.
- 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

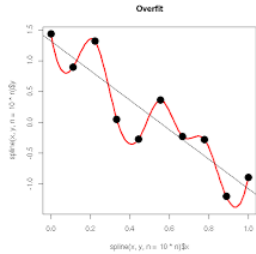
Food for thought: deep philosophical differences, but deeper connections (McCulloch/Pitts, AlphaGo)?

- 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, 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.

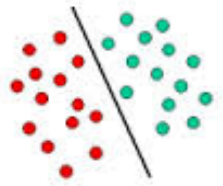
1801

- 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** (broadly construed) perspective is also how we will frame the topics in this course.

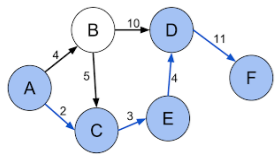
Early ideas from outside AI



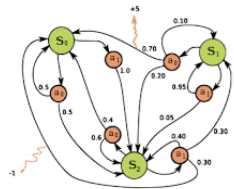
1801: linear regression (Gauss, Legendre)



1936: linear classification (Fisher)



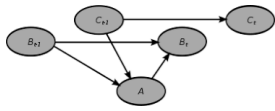
1956: Uniform cost search for shortest paths (Dijkstra)



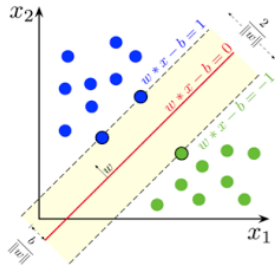
1957: Markov decision processes (Bellman)

- 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".
- Besides 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



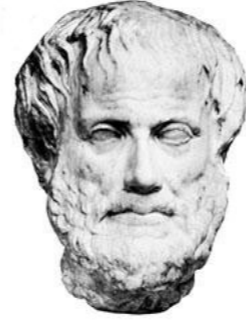
1985: Bayesian networks (Pearl)



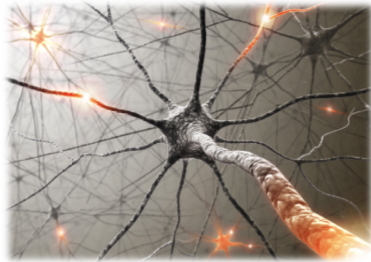
1995: Support vector machines (Cortes/Vapnik)

- 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.
- This is because for much of the 1990s and 2000s, the term AI wasn't actually used as much as it is today, partly to put distance between the most recent failed attempts in symbolic AI and partly because the goals were more down-to-earth.
- 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. SVMs were easier to tune than neural networks and became the favored tool in machine learning.

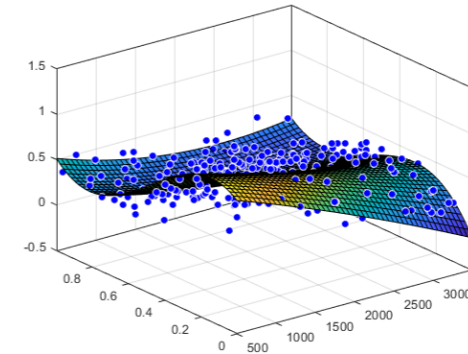
Three intellectual traditions



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, which with today's data and computing resources, has proven capable of conquering ambitious problems.
- Finally, **statistical AI** foremost offers mathematical rigor and clarity. For example, we define an objective function separate from the optimization algorithm, or have a language to talk about model complexity in learning. 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>



Roadmap

AI history

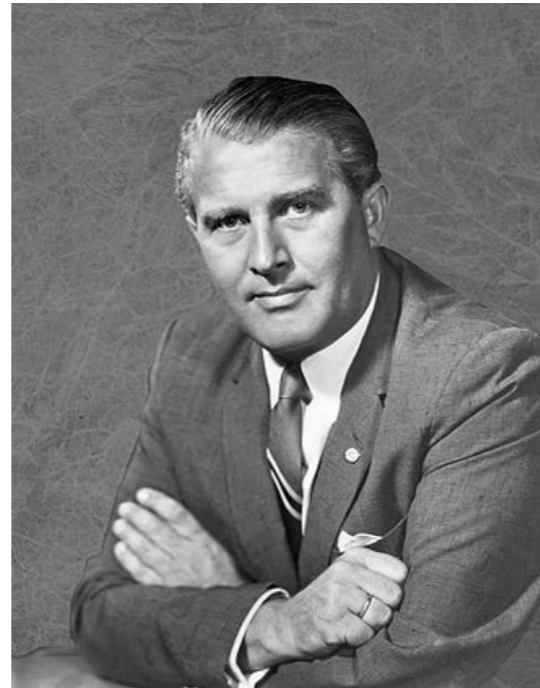
Ethics and responsibility

Course content

- Ethics and responsibility (we will use the terms interchangeably in this course) is a big, messy, and at times controversial topic. But it is essential that any researcher or practitioner of AI embrace responsibility as a top-of-mind consideration alongside the technical considerations.

Why care about responsibility?

Isn't technology value-neutral?



Wernher von Braun

*"Once the rockets are up,
Who cares where they come down?
That's not my department,"
Says Wernher von Braun.*

Lyrics: Tom Lehrer

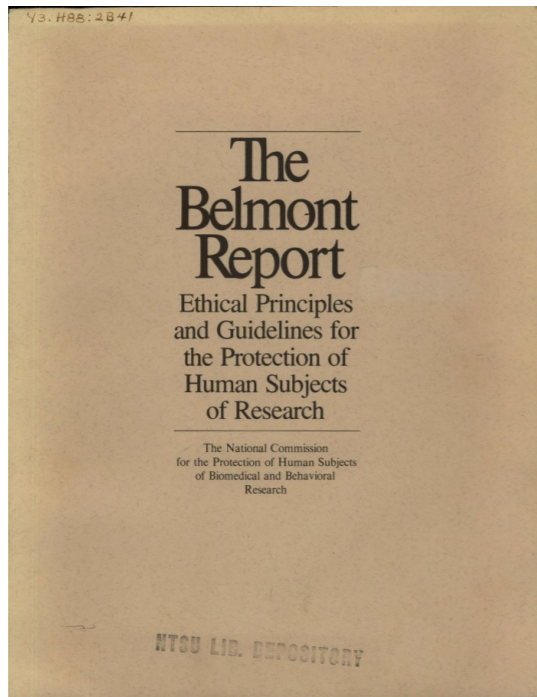


- The first question to ask ourselves is why we care. One might think that technology is value neutral: the goal of technologists should be to just develop the technology, and it is someone else's job to figure out how to make sure it's applied appropriately.
- That's what Wernher von Braun thought. He was a brilliant scientist interested in rocket technology, and he ended up joining the Nazi Party and helping Hitler develop rockets during World War II. Then he came to the United States to help with the space program. His attitude is captured aptly by Tom Lehrer's song.
- This extreme example illustrates the flaw with thinking that technology is value neutral, because it is impossible to separate the technology from its social context.

Goal of responsibility

Goal: ensure AI is developed to benefit and not harm society

High-level principles: respect for persons, don't do harm



ACM Code of Ethics and Professional Conduct

Preamble

Computing professionals' actions change the world. To act responsibly, they should reflect upon the wider impacts of their work, consistently supporting the public good. The ACM Code of Ethics and Professional Conduct ("the Code") expresses the conscience of the profession.

Microsoft AI principles

We put our responsible AI principles into practice through the Office of Responsible AI (ORA), the AI, Ethics, and Effects in Engineering and Research (Aether) Committee, and Responsible AI Strategy in Engineering (RAISE). The Aether Committee advises our leadership on the challenges and opportunities presented by AI innovations. ORA sets our rules and governance processes, working closely with teams across the company to enable the effort. RAISE is a team that enables the implementation of Microsoft responsible AI rules across engineering groups.

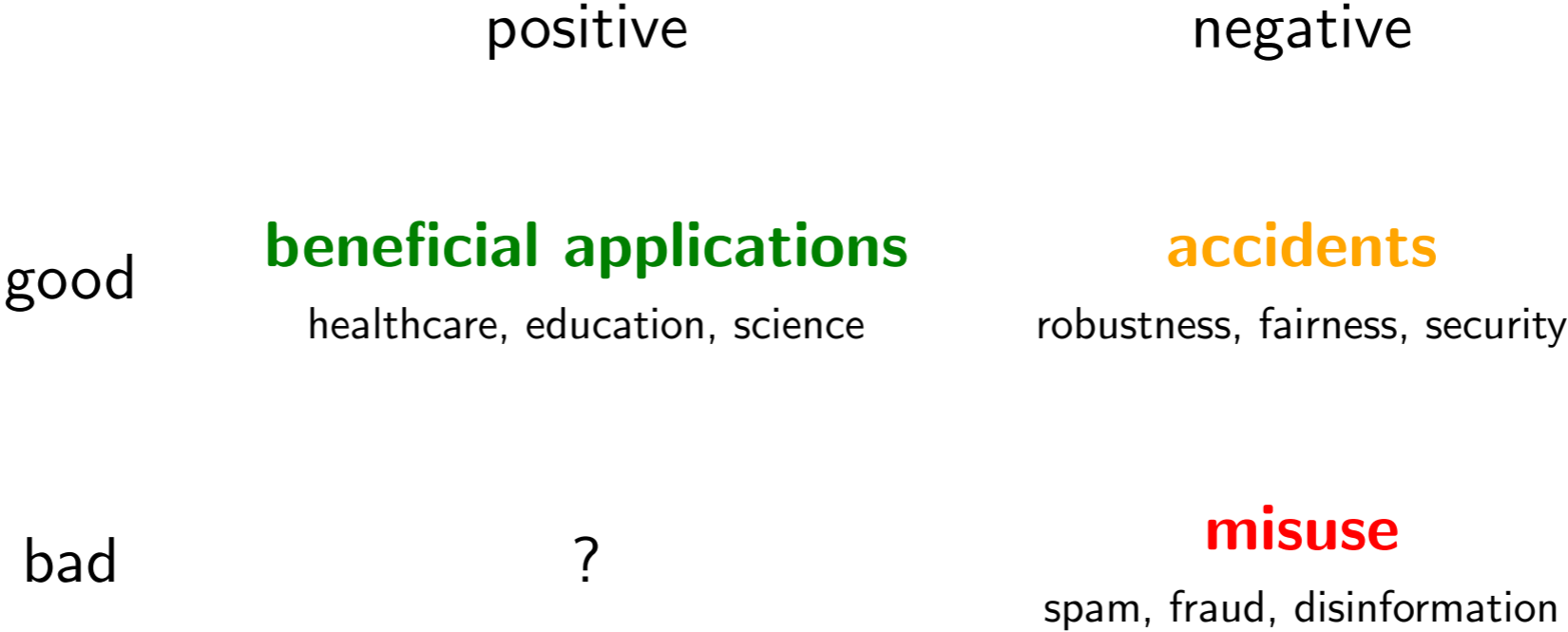
Key question: how to operationalize these principles?

- Responsibility is about ensuring that AI is developed in a way that benefits and doesn't harm society.
- What does this mean? We can appeal to high-level principles put forth by statements such as the Belmont Report from the 1970s, which laid the foundation for human subjects research, ACM Code of Ethics, and various responsible AI guidelines from industry.
- These principles are usually agreeable, but the key question is how do we operationalize these high-level principles?

Intent versus impact

← Impact →

↑ Intent ↓

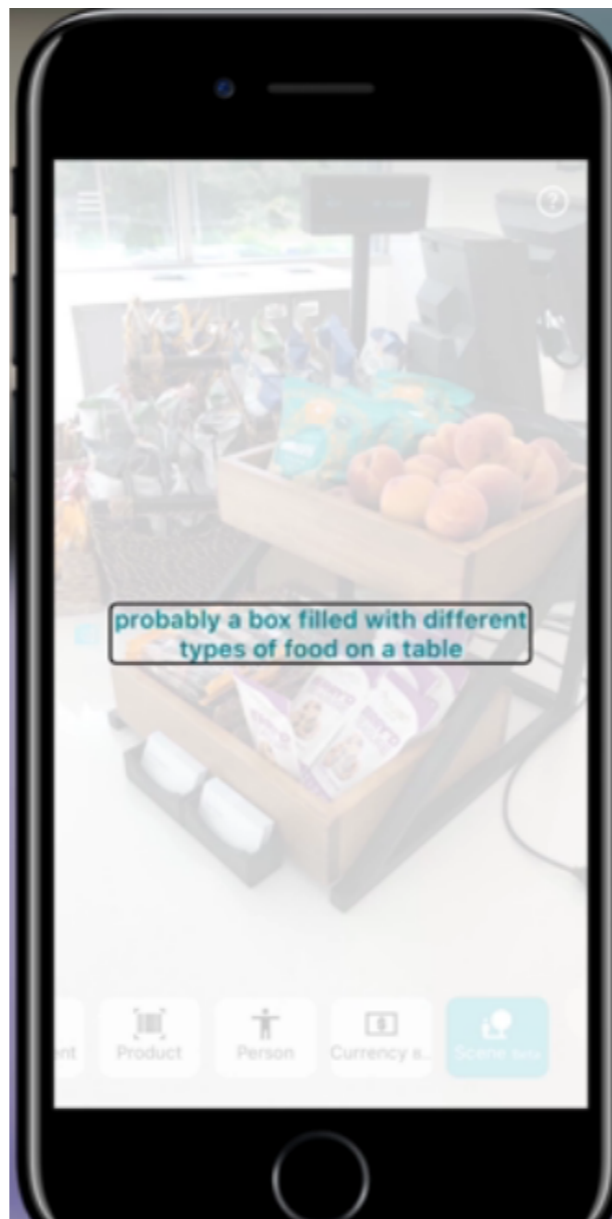


- Here is a framework that helps us think through the space of things that could go right or wrong.
- There are two axes: intent and impact, each of which could be good or bad.
- One could have good intentions that results in positive impact. This is the space of **beneficial applications** of AI. There are tons of areas where society could benefit from applying AI to areas of need: healthcare, education, access to justice, and science (biology, chemistry, physics, etc.).
- One could have bad intentions that result in negative impact. These examples of **misuse** include generating spam, performing fraud, generating disinformation.
- The third and more subtle category is **accidents**. These are cases where one has good intentions, but we still end up with negative impact. As we will see later, this happens due to the gap between the real world and the model.
- Finally, the case where one has bad intentions and still ends up with positive impact is exceptionally rare.

Beneficial applications

- Here are a simple set of examples of how AI could be used to benefit people.

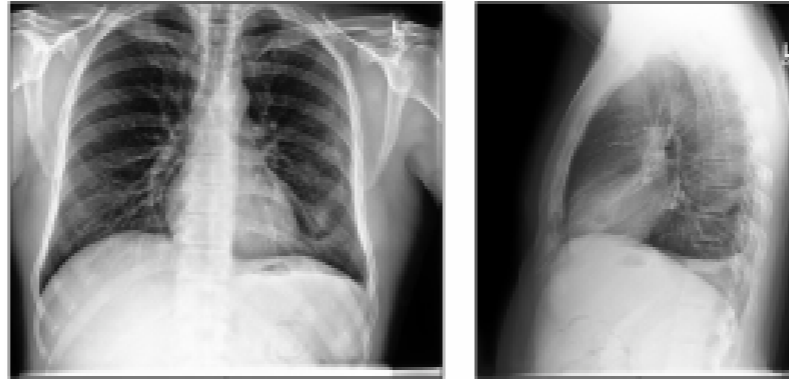
Visual assistive technology



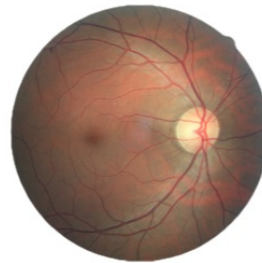
- This example is the Seeing AI app from Microsoft Research, which narrates whatever the camera is pointed at.
- This visual assistive technology could be a game-changer for the visually impaired.
- Conversely, auto-captioning technology, which turns sound into sight, is potentially also quite useful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

Healthcare

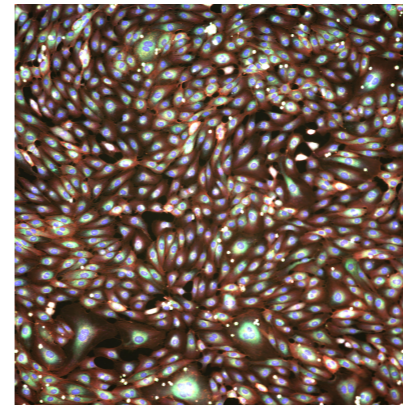
Chest radiology



Diabetic retinopathy

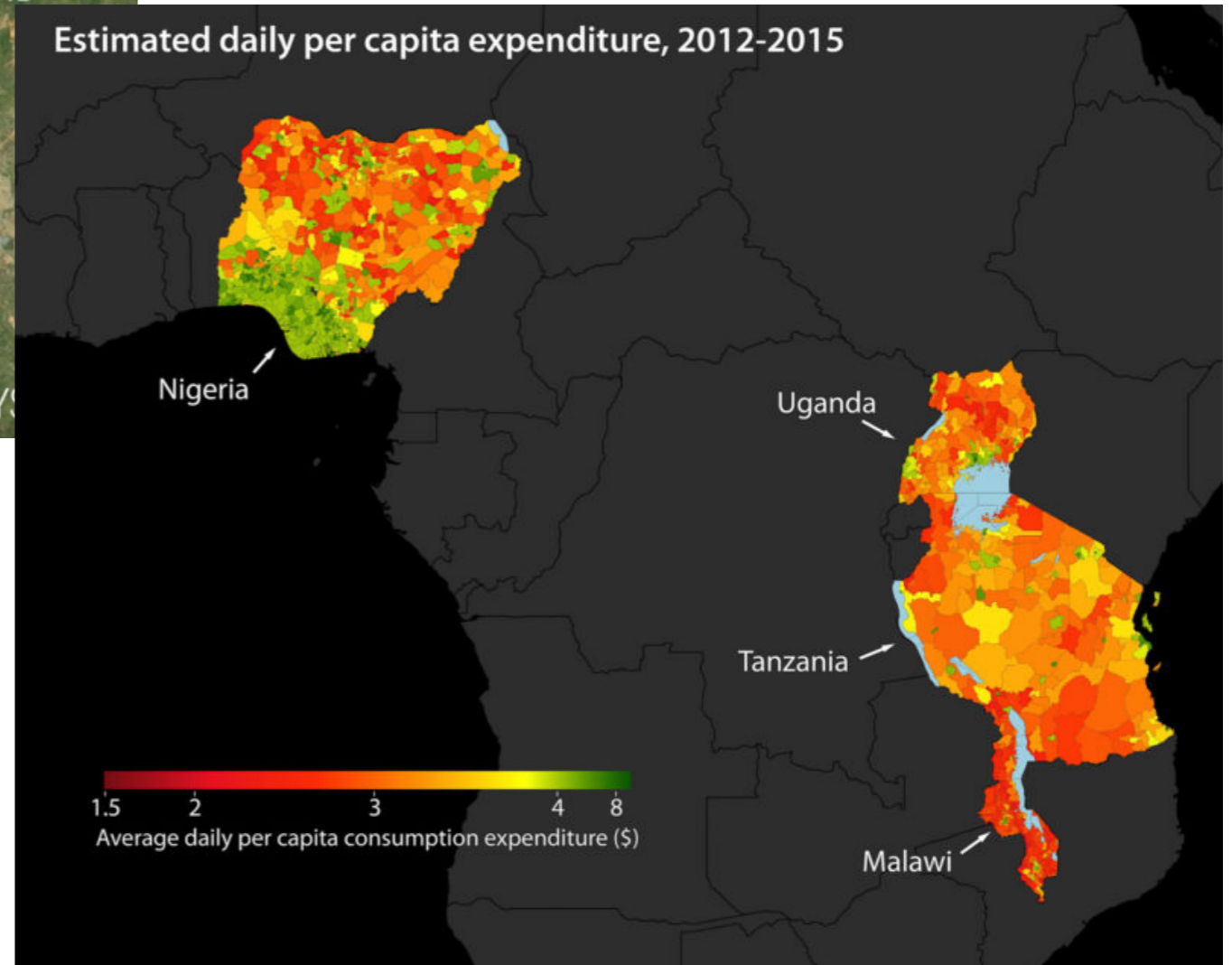
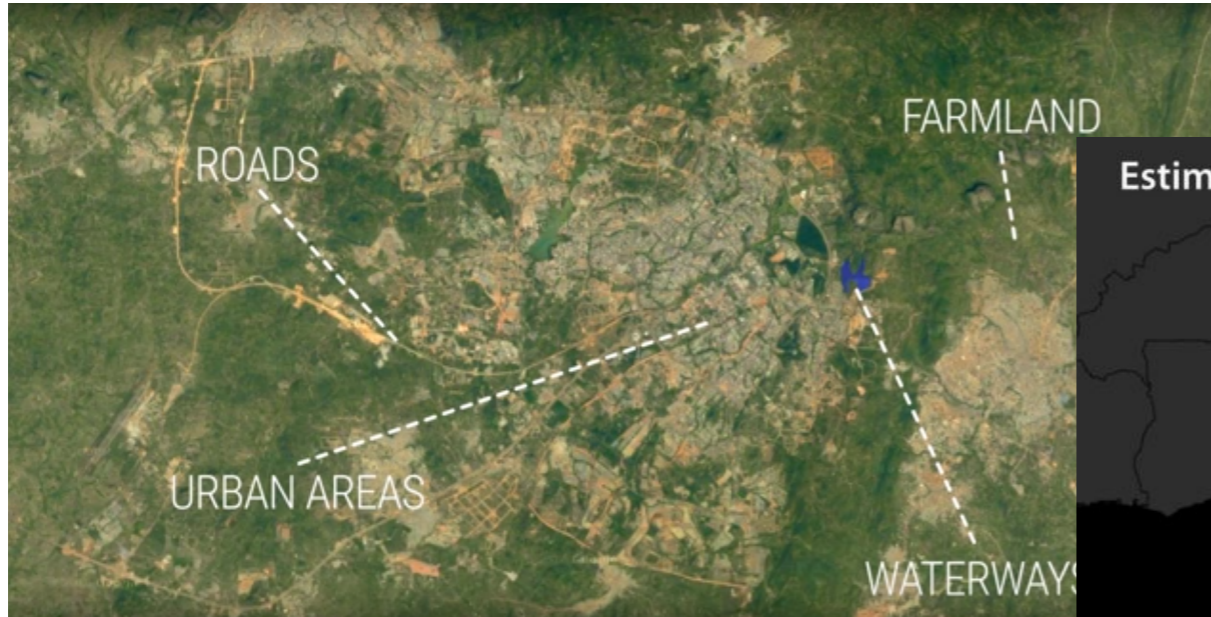


Drug screening for COVID-19



- 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.

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.

Misuse

- Now let us think about where AI could potentially have negative impact (in other words, be misused).

Disinformation



Others: spam, fraud, personal attacks

- On the left is a video that shows Obama saying things he never said. On the right you see Stable Diffusion, a new text-to-image model that can be used to generate photorealistic fictitious scenes.
- 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.

Dual-use technology

Definition: a dual use technology is one that can be used both to **benefit** and to **harm**.

Examples:

rockets

nuclear power

gene editing

social networks

AI

- You might be thinking: well, I would never misuse AI! However, it's not so simple because of the very nature of AI: it is a dual use technology, which is something that can be either used for good or for evil.
- There are many other examples of dual use technology, each very powerful in their own right. They could be used to create energy, to cure diseases, to connect people, but they also could be weaponized.

Levels of abstraction

deep learning

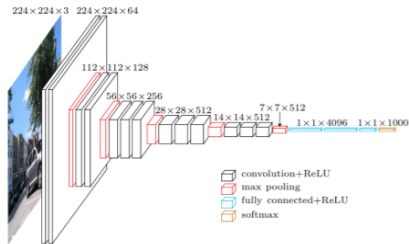


image generation



deepfakes



autonomous weapons



generality

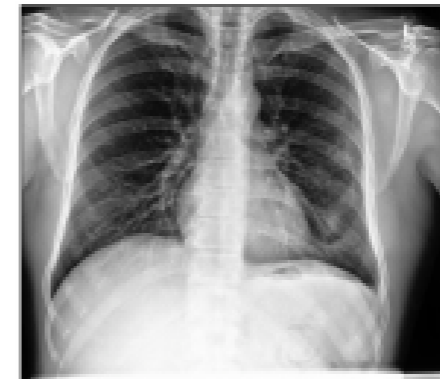
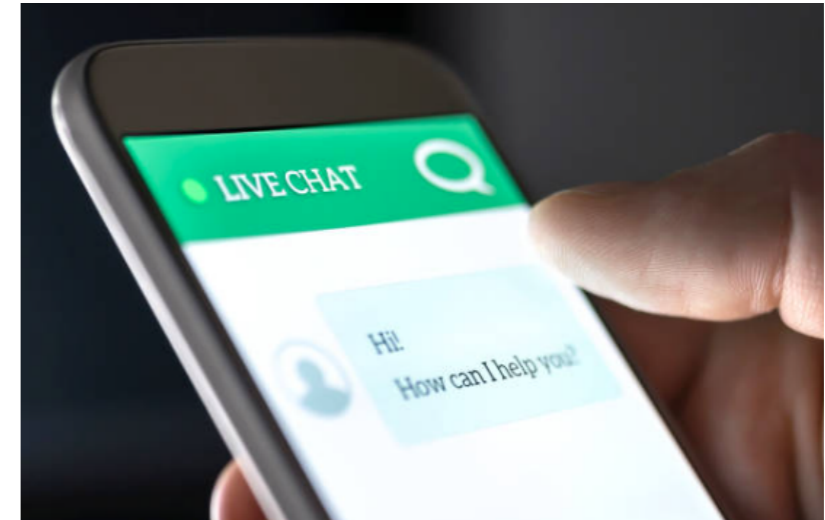
specificity

- The type of impact that an AI researcher or developer has depends on the level of abstraction they are working at.
- At the most specific end of the spectrum, we can consider concrete use cases. For example, autonomous weapon systems, which are a very controversial technology, are explicitly created to kill and destroy. Here it's relatively easy to see the potential harms.
- Deepfakes are also ethically contentious, though arguably less so. While they have genuine use cases in entertainment, they do have serious negative consequences for the integrity of our information ecosystem: malicious actors will be able to sow confusion by disseminating disinformation, and people will no longer be able to distinguish truth from fiction.
- If generating faces is problematic, how about generating images (e.g., dogs)? At the surface, this seems harmless, but a lot of research in this area improves the overall capabilities of generative models, which enable deepfakes, but can also be used to perform data augmentation to improve the accuracy and robustness of any machine learning system.
- Pushing this one step further, all of these applications are made possible by advances in deep learning. If a researcher comes up with a new model architecture that is enabling, are they responsible for its downstream consequences?
- The higher upstream you go, they more diffuse your impact, but you still have impact.

Accidents

- The final category are accidents, or unintended consequences, where one has good intentions but ends up having negative impacts.

Complex real-world problems



- Recall that the goal of AI is to develop the machinery to tackle complex real-world problems.

- In this class, we will emphasize the value of trying to create mathematical abstractions of the real-world (i.e., models) in order to make technical progress.
- However, AI systems don't live in the mathematical world. When deployed, they affect real people and have real consequences.
- So we need to understand those consequences and be constantly mindful of the gaps introduced by our assumptions.

Optimizing the wrong objective function



Misalignment between real-world objective and system's objective

- Here is an example of a reinforcement learning agent who has been trained to play a video game, where the goal is to race a boat around a course.
- Except for the goal (that the system is given) isn't to race a boat around a course; rather, it is to maximize the number of points. So by optimizing for the number of points, the agent has learned to repeatedly loop around in the lagoon hitting the same targets and racking up points.
- This example is an instance of **reward hacking** and shows that the difference between the real-world objective (which might be finishing the race) and the objective function given to the AI could cause behavior that is quite unanticipated.

Optimizing the wrong objective function












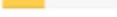






Is maximizing clicks a good objective function?

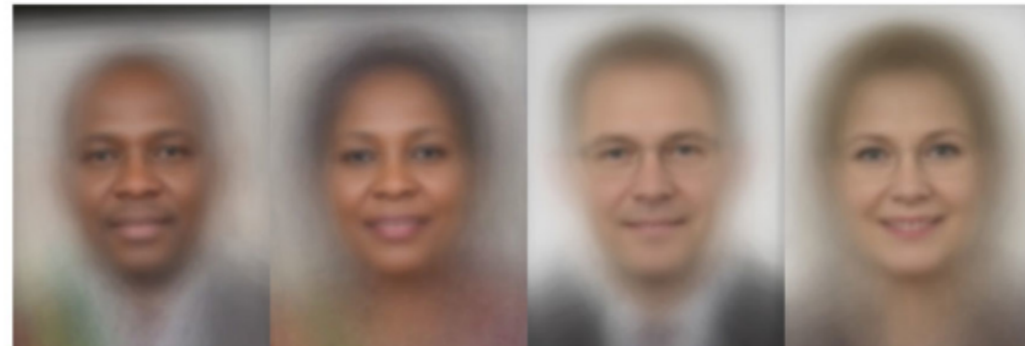


Beware of surrogates and mis-aligned incentives

- In general, optimization is a powerful paradigm: it allows you to express a desire (in the form of an objective function) and then put resources behind it to make it come true.
- However, the big question is what the objective function should be? Ideally it would be something like happiness or productivity, but these things are impossible to measure, so often **surrogates** (approximations) are used.
- Moreover, businesses are **incentivized** to maximize profit, which is not always aligned with what's good for people.
- For example, most Internet companies use clicks or views as a major component of their objective functions. But people's reflexive actions are not representative of their long-term goals. At a societal level, we have seen that this leads to big problems such as polarization.

Fairness: performance disparities

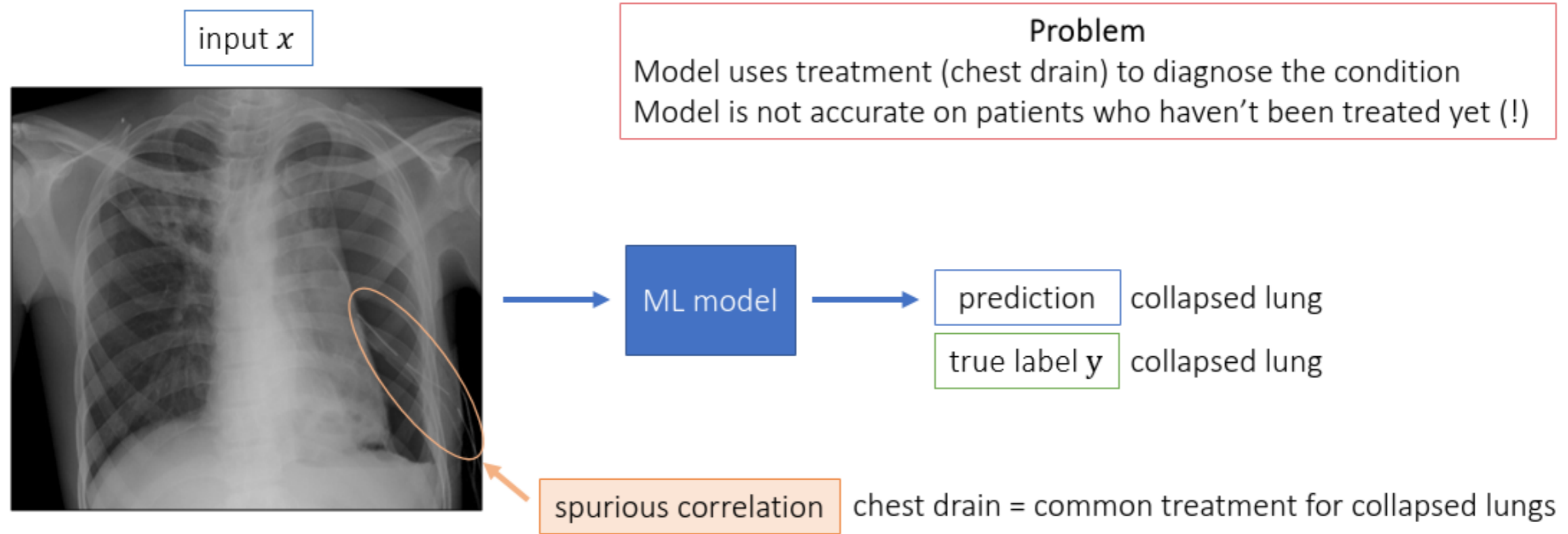
Gender Classifier	Darker Male	Darker Female	Lighter Male	Lighter Female	Largest Gap
 Microsoft	94.0% 	79.2% 	100% 	98.3% 	20.8% 
 FACE++	99.3% 	65.5% 	99.2% 	94.0% 	33.8% 
 IBM	88.0% 	65.3% 	99.7% 	92.9% 	34.4% 



Inequalities arise in machine learning

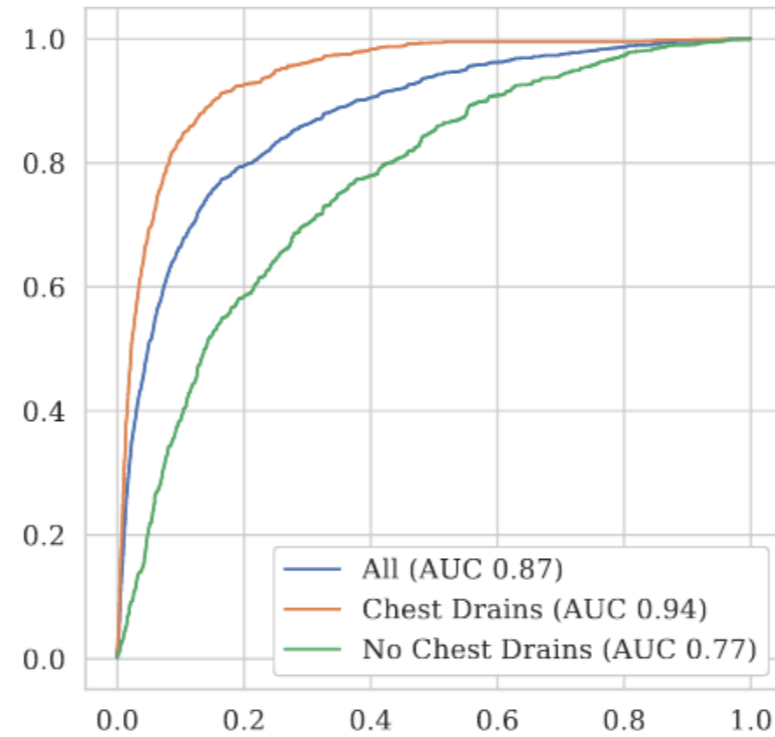
- GenderShades is a famous study that shows that standard classifiers can work poorly on certain groups within the population. Usually, this is due to lack of representation in the data.
- One can alleviate this problem by collecting more data for under-representative segments of the population. But this can be hard and expensive to do, and companies might not be incentivized to invest in this unless regulation changes.
- A complementary solution is to minimize the maximum group loss, which embodies John Rawls's difference principle of helping the worst-off. Technical fixes that don't involve gathering more data often come with tradeoffs such as slightly decreased performance for other groups. How to address the tradeoffs is a philosophically difficult question, the answer to which may vary depending on the setting and stakes of the classification task.
- In all cases, **auditing** is a powerful force, to increase transparency, and drive change. For example, after the Gender Shades project showed performance disparities, all the companies went and significantly closed the performance gaps.

Robustness: spurious correlations



- Take the task of predicting whether a chest x-ray is indicative of collapsed lung.
- Apply standard convnet machinery from computer vision and it works reasonably well. But take a closer look: see that thin tube coming out?
- This is a chest drain, which is a common treatment for a collapsed lung. And it turns out this is one of the signals that the model is picking up on.

Robustness: spurious correlations



Subpopulation of untreated patients are worse off than treated patients!

- This means that patients with chest drains obtain much higher AUC than patients without. But wait a minute! The patients without chest drains are exactly the subpopulation of untreated patients, who we most care about making accurate predictions, and they're the ones that suffer.
- Many of these issues are due to the fact that machine learning thrives on complex models fitting correlations in data.

Security

[Evtimov+ 2017]



[Sharif+ 2016]

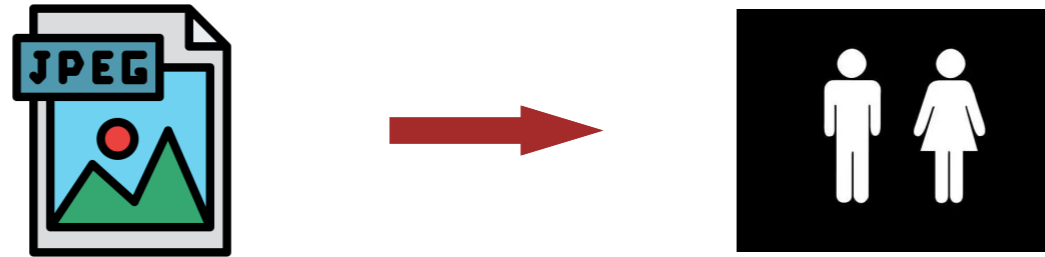


Adversaries at test time!

- 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.

Task definition

Gender classification:



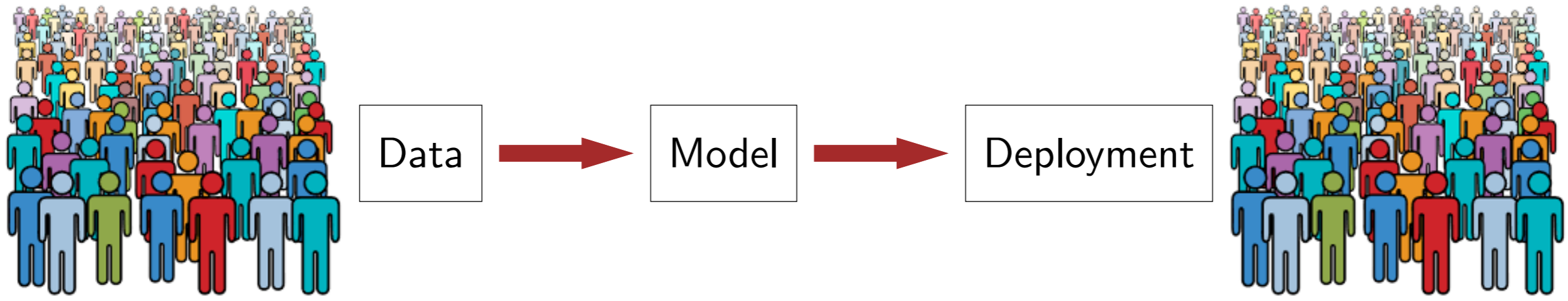
Questions:

- Is this a meaningful task given the inputs? Self-identification?
- Is the output space meaningful? Other genders?

Always think about the task setup

- Then there are fundamental issues that stem from its very definition of a task regardless of how you choose to tackle it. As an example, consider gender classification from an image. There are two issues here.
- First, is this a meaningful task given the **inputs**? Always remember that the inputs given to a machine learning algorithm is an approximation made by the dataset creator: it was taken out of context and put into a dataset. If you are interested in gender being defined by self-identification, then the physical appearance distilled down into a still image might not be appropriate.
- Second, is the **output space** meaningful? Machine learning classification is fundamentally about categorizing the complex real-world into a convenient discrete set of categories. Inevitably, this categorization will be imperfect. Now the question is who gets marginalized or excluded by this categorization and what are the harms?
- The lesson is to always think about the task itself in the context of the real-world, before even attempting to solve the task.

Two contact points



privacy
consent
copyright
jobs

alignment
fairness
robustness
security
definition

feedback loops

- So far we've looked at how a model that's deployed could have impact on people.
- This is not the only way that machine learning impacts people. Models are trained from data, and data comes from people. So a second class of issues to worry about is the impact due to data collection.

Data

- Web-scraped data can contain offensive content, historical biases

MIT takes down 80 Million Tiny Images data set due to racist and offensive content



- Consent: Should a datum (e.g. a picture of my dog) whose owner or creator intended it for one use be allowed to be used in another application (e.g. scene classification) without permission?

- Recall that any machine learning (which powers most AI systems) depends on data, so we must question what is in the data.
- TinyImages was a dataset of 80 million images collected in 2006 based on WordNet + scraping the Internet. It was taken down in July 2020, because it was found that some of the categories were derogatory and offensive.
- GPT-3 was trained on text scraped from the Internet, which clearly has a lot of offensive, problematic content.
- In general, since predictions of machine learning models reflects the training data, using a uncurated web scrape can lead to unpredictable harms, even if there was no ill intent.
- There is also the question of whether data produced for one purpose (e.g., photos I took to share with my friends) should be used for another purpose (e.g., building scene classification systems for self-driving cars) without consent, compensation, or even notification.

Data

How to
Stop Silicon Valley
from Building a
New Global Underclass

GHOST

Mary L. Gray and Siddharth Suri

WORK

Data is produced by human labor

- When one thinks of AI, one thinks of the technology, Because of our focus on the technology, we often have the impression that the introduction of AI always reduces human labor and makes things more efficient. However, AI is not free and requires resources.
- Ghost Work documents the immense and often invisible human labor (crowdsourcing) that is crucial for making AI, such as labeling data or moderating flagged content and how crowdsourcing platforms create a new class of unstable gig-economy labor.
- As another example, machine learning practitioners draw a sharp distinction between labeled data (expensive to obtain) and unlabeled data (cheap or even free to obtain), where the latter is exemplified by web scrapes. However, if you think about it, all data is created by people expending capital. Unlabeled data such as "raw text" (books and articles) actually took substantial time and effort to produce. It's only free because the machine learning developer is not paying for the value of the asset.

Automation and jobs



- Text-to-image models (e.g., DALL-E) can replace jobs?
- Models are actually trained on the labor of the artists

- Recently, text-to-image models such as OpenAI's DALL-E or Stability AI's stable diffusion model have wowed the world with its stunning generations. They have even been used to win art contests.
- However, many artists are outraged: First, if anyone who can mumble a few words can generate art that takes years of training to do manually, there could be a direct threat to an artist's livelihood.
- Second, and perhaps even more infuriating is the fact that these models were trained on millions of artists' work, and there was no consent nor compensation for using that work as training data.

What should we do?

- Given all these weighty issues, what should we do?

Transparency

Model Cards for Model Reporting

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, Timnit Gebru
{mmitchellai,simonewu,andrewzaldivar,parkerbarnes,lucyvasserman,benhutch,espitzer,tgebru}@google.com
deborah.raji@mail.utoronto.ca

Datasheets for Datasets

TIMNIT GEBRU, Black in AI
JAMIE MORGENSTERN, University of Washington
BRIANA VECCHIONE, Cornell University
JENNIFER WORTMAN VAUGHAN, Microsoft Research
HANNA WALLACH, Microsoft Research
HAL DAUMÉ III, Microsoft Research; University of Maryland
KATE CRAWFORD, Microsoft Research

Document potential issues

- All of these ethical and responsible AI issues are extremely complex, and involve tradeoffs. There are no simple solutions.
- But one of the key tenets is **transparency**, a necessary but not sufficient condition for responsibility. Increasingly, there is efforts such as model cards or datasheets that encourage the community to at least acknowledge and document any deficiencies of models and datasets, to declare the intended and prohibited uses, to provide a mechanism for reporting problems, etc.

Choosing problems

- **Beneficial applications:** work on directly benefiting society
- **Human-in-the-loop:** augment humans, not replace them
- **Differential privacy:** protect individual liberty
- **Few-shot learning:** open up applications with little data

- The idea that technology is value-neutral is simply a refusal to confront reality. There are tons of ways in which an AI researcher or developer's concrete action can have a meaningful impact on the direction of the field.
- The most obvious one is work on beneficial applications. We are here working at the specific level of abstraction, where the real-world impact can be more easily controlled and monitored.
- But we can also work on general-purpose techniques that orient the field. For example, working more human-in-the-loop systems can help us avoid job loss. Differential privacy has the potential to protect the privacy of individuals (although this needs to be done with care lest we worsen the privacy). And you might want to work on few-shot learning, which can help people who have little data (e.g., low-resource languages).



Summary

- Any technology is value-laden
- AI is a dual use technology (could benefit or harm)
- Intent x impact: beneficial applications, misuse, accidents
- Accidents stem from gaps between the real-world and model
- Responsibility: no correct answers, many tradeoffs, something to always keep in mind

- To summarize, AI is value-laden; your choices do have an impact on AI develops.
- AI, like any dual-use technology, is an amplifier: it can lead to both very good and very bad outcomes. Even if you are working at a higher-level of abstraction, you still have an impact (positive or negative), though it might be harder to see.
- We discussed the types of impact: the easy cases are beneficial applications and misuse. The more nuanced category is accidents, which arise mostly because AI operates on models, which might differ from the real world.
- Finally, responsibility is a complex topic and there are no easy answers. At some level, it is more important to engage in the process of debate and reflection, rather than having an algorithm or recipe to blindly execute.



Roadmap

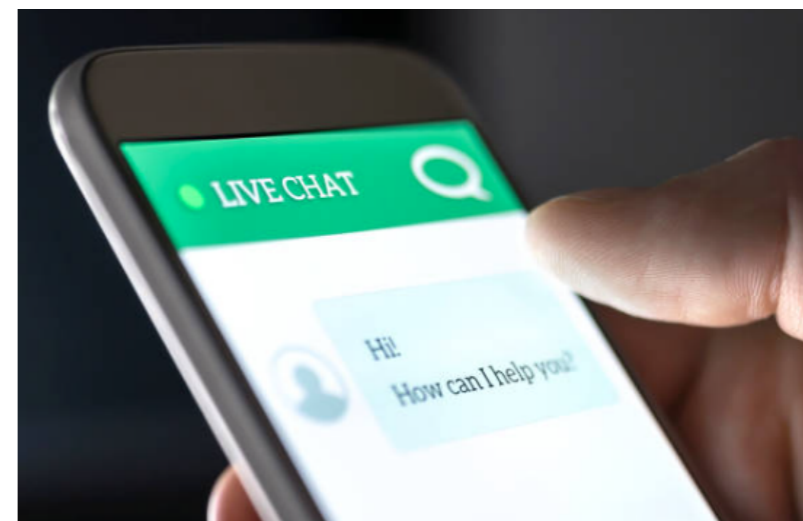
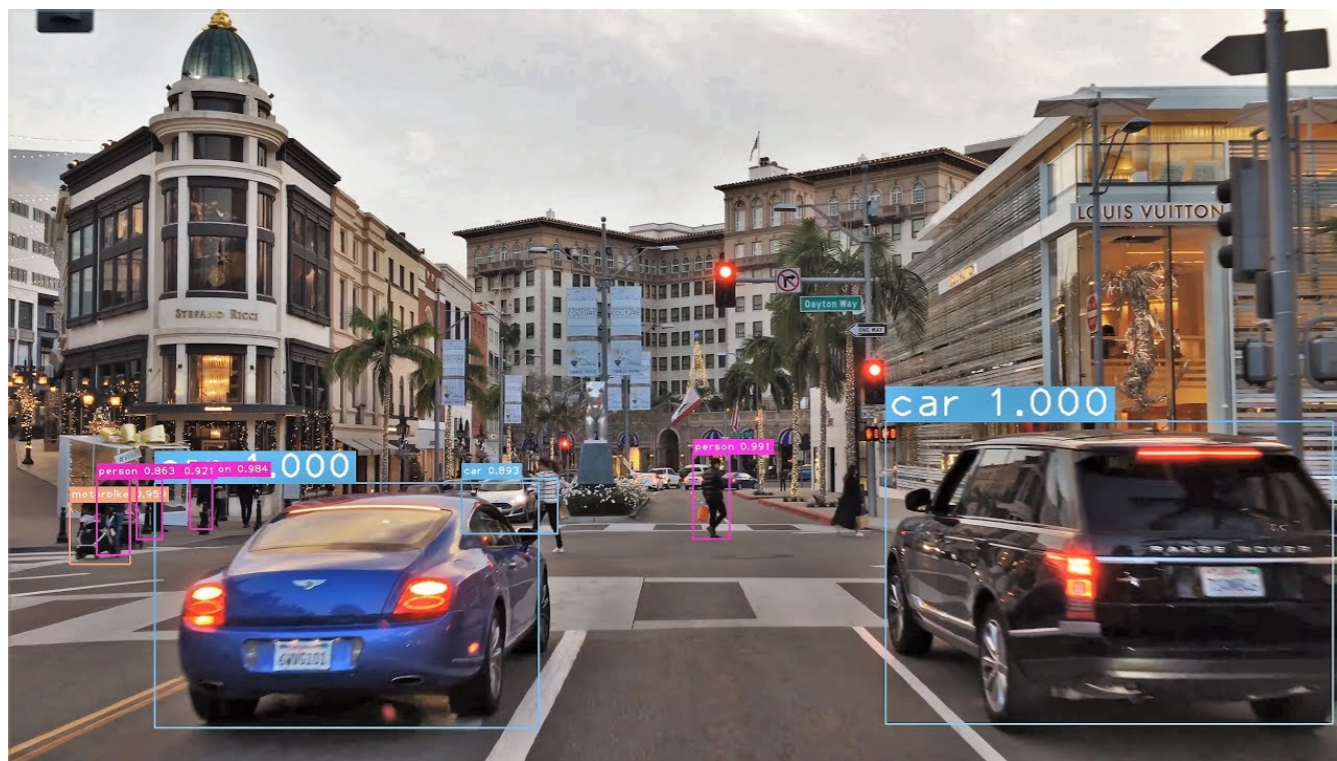
AI history

Ethics and responsibility

Course content

- Let's now talk about what we will cover in this course. In this module, we'll cover the modeling-inference-learning paradigm, which will help us tackle complex real-world problems. Within this paradigm, we'll survey the different types of methods that we will encounter in this course.

Complex real-world problems



- AI is about using technology to tackle complex real-world problems.
- What are some complex real-world problems that come to mind?
- Self-driving cars have to sense the complex scene in front of them (in real-time!) and make safe decisions that advance the car towards its destination.
- Virtual assistants have to understand the rich subtleties of natural language, infer the user intent, and generate appropriate responses.
- Medical systems have to interpret the subtle, noisy cues in medical images, take into account the messy incomplete patient information, and generate reliable diagnoses that can be trusted by doctors.
- These are really really hard problems. How do you even begin to tackle them?

Bridging the gap



?

```
# Data structure for supporting uniform cost search.
class PriorityQueue:
    def __init__(self):
        self.time = 0.0000
        self.heap = []
        self.priorities = {} # Map from state to priority

    # Insert (state) into the heap with priority (newPriority) if
    # (state) isn't in the heap or (newPriority) is smaller than the existing
    # priority.
    # Return whether the priority queue was updated.
    def update(self, state, newPriority):
        oldPriority = self.priorities.get(state)
        if oldPriority is None or newPriority < oldPriority:
            self.priorities[state] = newPriority
            heappush(self.heap, (newPriority, state))
            return True
        return False

    # Return (state with minimum priority, priority)
    # or (None, None) if the priority queue is empty.
    def removeMin(self):
        while len(self.heap) > 0:
            priority, state = heappop(self.heap)
            if self.priorities[state] == self.time: continue # Duplicated priority, skip
            self.priorities[state] = self.time
            return (state, priority)
        return (None, None) # Nothing left...

=====
# Simple examples of search problems to test your code for Problem 1.

# A simple search problem as the number line.
# From 0 to 10, want to go to 10, costs 1 to move down, 2 to move up.
class NumberLineSearchProblem:
    def startState(self): return 0
    def isGoal(self, state): return state == 10
    def successors(self, state): return [(('west', state-1, 1), ('east', state+1, 2))]

# Function to create search problems from a graph.
# You can use this to test your algorithm.
def createSearchProblemFromGraph(start, goal, description):
    # Parse the graph
    graph = collections.defaultdict(list)
    for line in description.split("\n"):
        if line.startswith('#') or line.startswith('E'): continue
        # Edge from state a to state b.
        a, b, cost = line.split(" ")
        cost = float(cost)
        # Add an in the same as the destination state (b).
        graph[a].append((b, cost))
```

- At the end of the day, we need to write some code (and possibly build some hardware, too).
- But there is a huge chasm between the problem and the solution.
- You don't want to just start jumping in and writing code without a clear plan of attack.

Paradigm

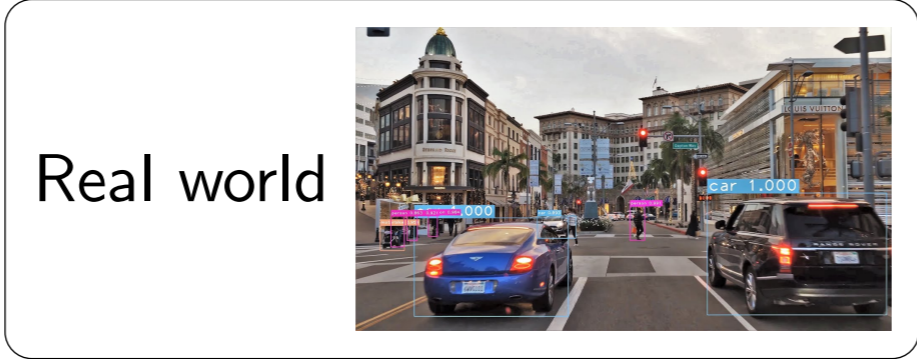
Modeling

Inference

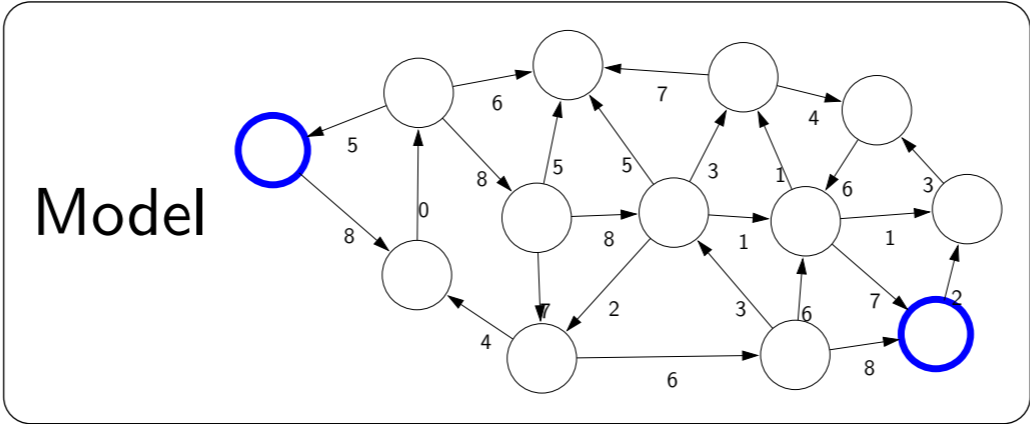
Learning

- So here is the plan of attack. We will adopt the **modeling-inference-learning** paradigm to help us navigate the solution space.
- There are three pillars, modeling, inference, and learning.
- In reality, the lines between the three pillars are blurry, but this paradigm serves as an ideal and a useful guiding principle.

Paradigm: modeling

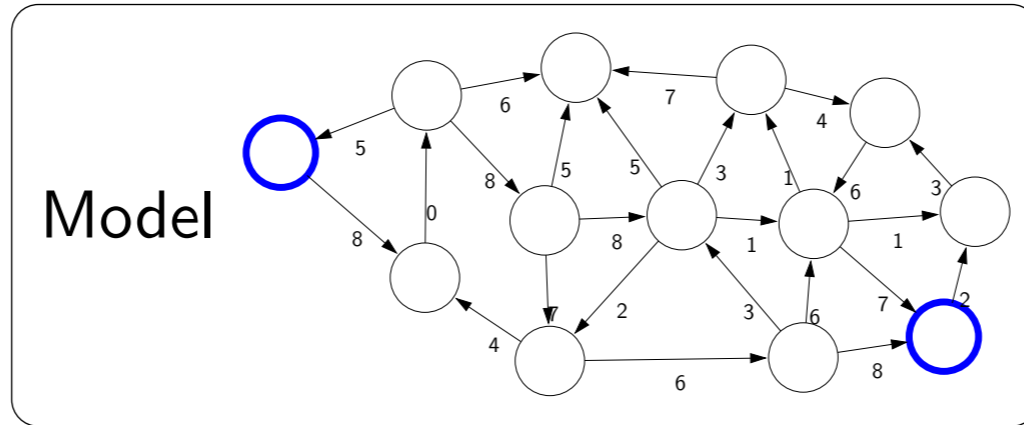


Modeling

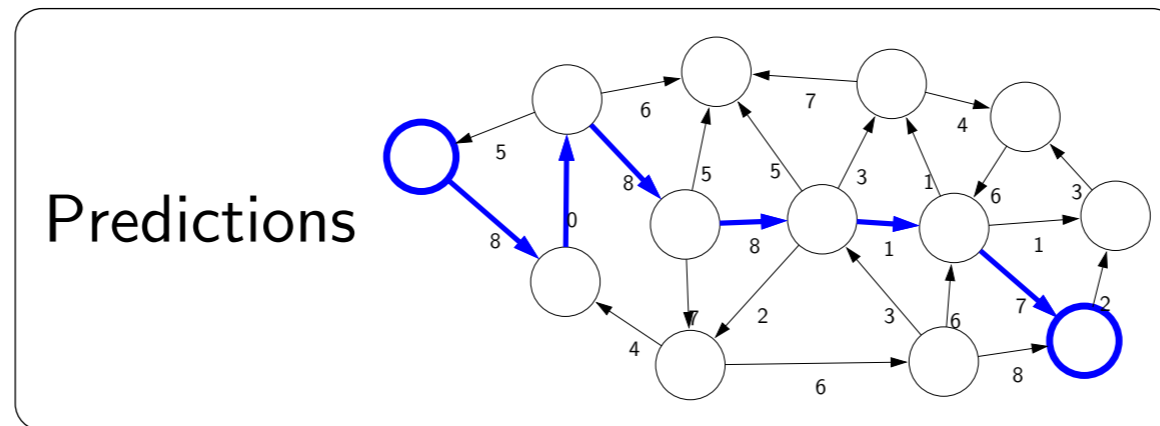


- The first pillar is modeling.
- Modeling takes messy real world problems and packages them into neat formal mathematical objects called **models**, which can be subject to rigorous analysis and can be operated on by computers.
- However, modeling is lossy: not all of the richness of the real world can be captured, and therefore there is an art of modeling: what does one keep versus ignore?
- (An exception to this are games such as Chess, Go, or Sudoku, where the real world is identical to the model.)
- We might formulate the driving problem as a route finding problem as a graph where nodes represent points in the city, edges represent the roads, and the cost of an edge represents the traffic on that road.
- If we do this, all the complexities of perception are ignored. Alternatively, we could make a model that only looks at perception and ignores planning.

Paradigm: inference



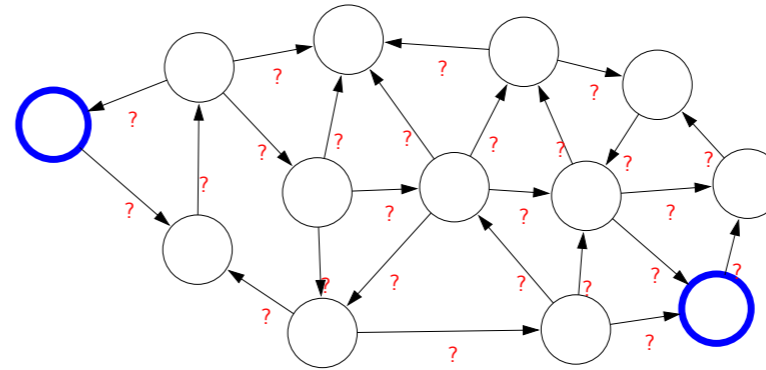
Inference



- The second pillar is inference. Given a model, the task of **inference** is to answer questions with respect to the model. For example, given the model of the city, one could ask questions such as: what is the shortest path? what is the 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

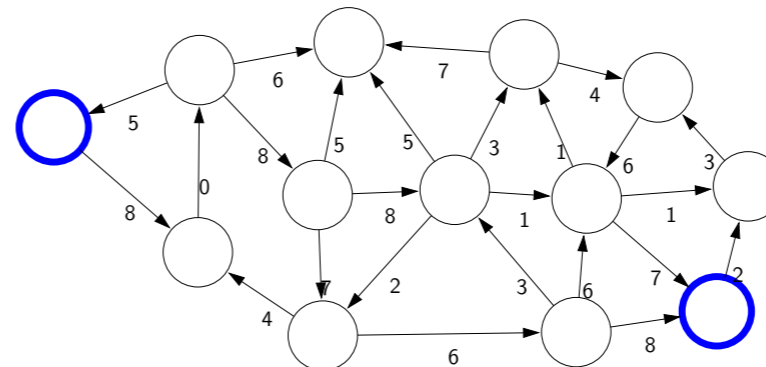
Model without parameters



+data

Learning

Model with parameters



- But where does the model come from? Remember that the real world is rich, so if the model is to be faithful, the model has to be rich as well. But we can't possibly write down such a rich model manually.
- The idea behind (machine) **learning** is to instead get it from data. Instead of constructing a model, one constructs a skeleton of a model (more precisely, a model family), which is a model without parameters. And then if we have the appropriate data, we can run a machine learning algorithm to tune the parameters of the model.
- Many of you are probably only seen models in the context of machine learning (e.g., neural networks), but I want you all to take a very broad view of what a model is. The idea of learning is not tied to a particular model family (e.g., neural networks). Rather, it is more of a philosophy of how to produce models.

Paradigm

Modeling

Inference

Learning

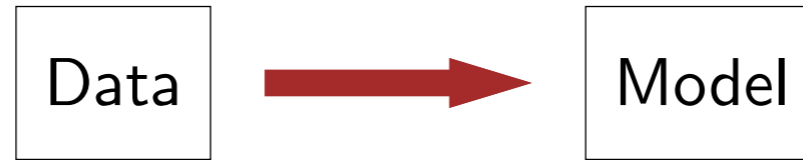
- To summarize, modeling simplifies the real world, inference answers questions against the model, and learning constructs the model from data.
- Note that each step could be challenging and require approximations.

Course plan



- Now in this course, we go through different types of models for representing different types of real-world problems.
- We start with models that are low-level (which just take an input and return an output — these are the models that you might be more familiar with in machine learning). Gradually, we progress to high-level models that are based on more logical reasoning.
- Before we start, we talk about machine learning, which can support all models.

Machine learning



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

- Supporting all of these models is **machine learning**, which has been arguably the most crucial ingredient powering recent successes in AI. From a practical perspective, machine learning allows us to shift the complexity of the model from code to data; instead of writing a lot of code to describe the model, we instead write a very simple model family (e.g., choice of neural network architecture) and focus on collecting the data to train a model within that family.
- The main conceptually magical part of learning is that if done properly, the trained model will be able to produce good predictions beyond the set of training examples. This leap of faith is called **generalization**, and is, explicitly or implicitly, at the heart of any machine learning algorithm. This can be formalized using tools from probability and statistical learning theory.

Course plan



- We now start our tour of models with reflex models.

What is this animal?

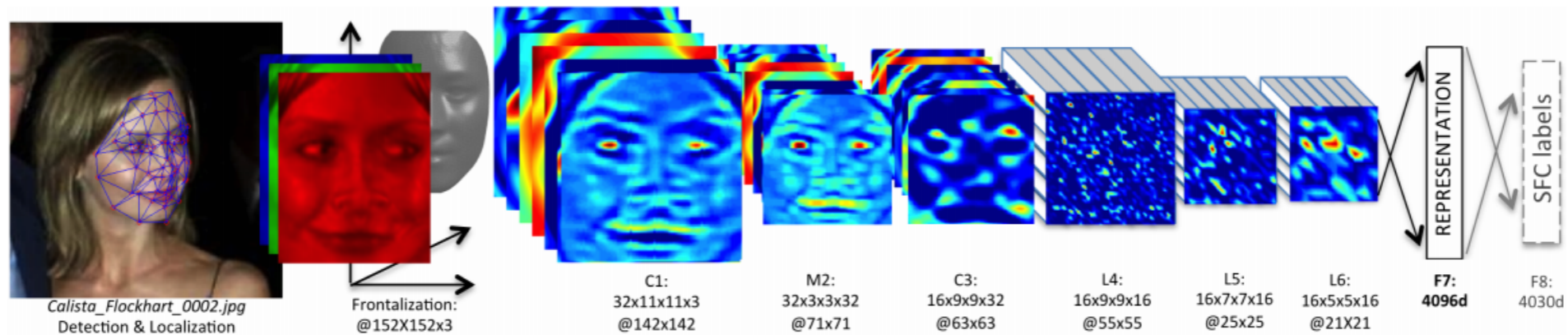


- Most of you could probably recognize the zebra in that split second.



Reflex-based models

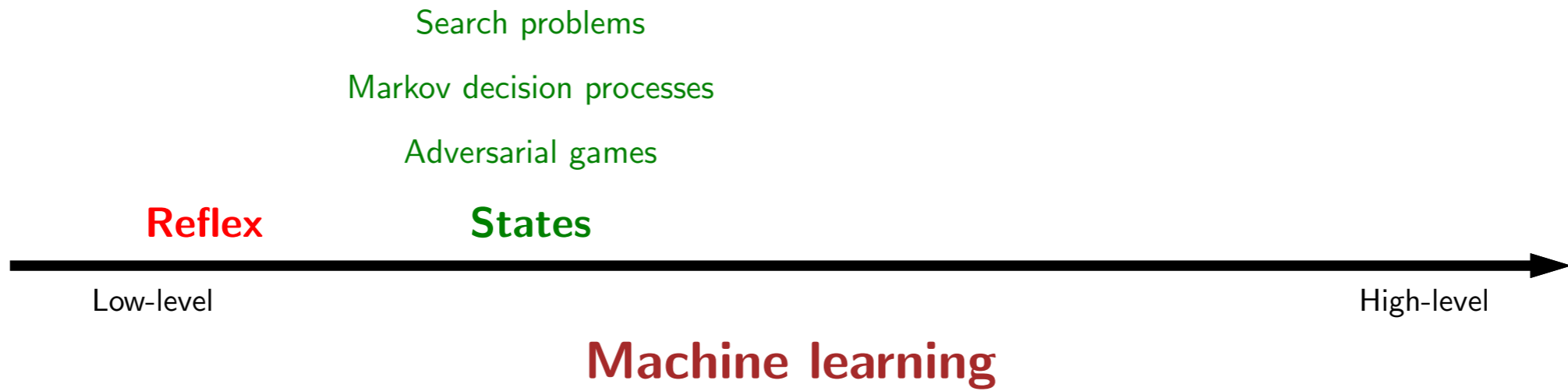
- Examples: linear classifiers, deep neural networks



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

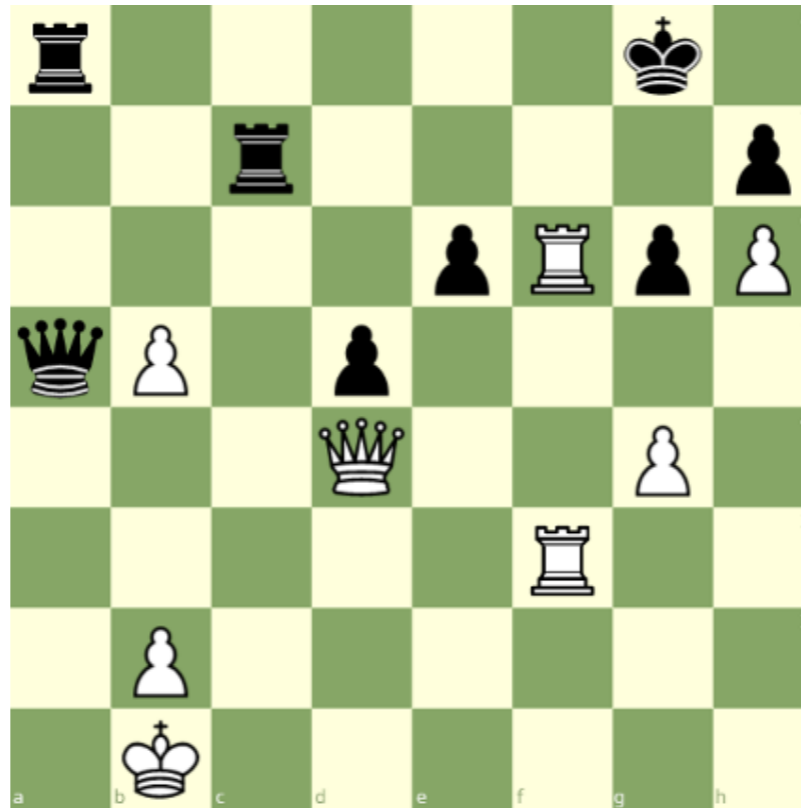
- A reflex-based model simply performs a fixed sequence of computations on a given input. Examples include most models found in machine learning, from simple linear classifiers to deep neural networks.
- The main characteristic of reflex-based models is that their computations are feed-forward; one doesn't backtrack and consider alternative computations.
- Inference is straightforward in these models because it is just running the fixed computations, which makes these models appealing.

Course plan



- Next, we will consider state-based models.

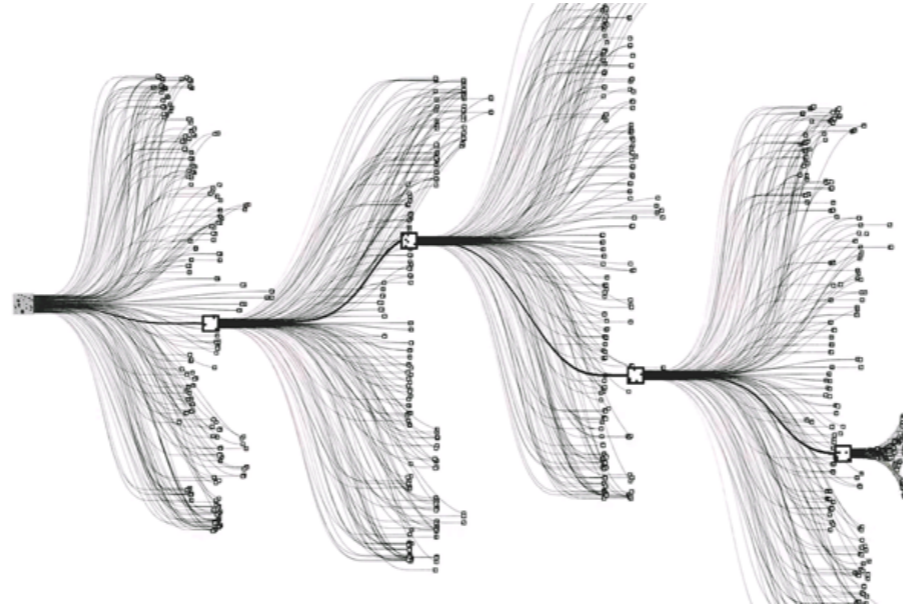
State-based models



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 more challenging than recognizing the zebra.

State-based models



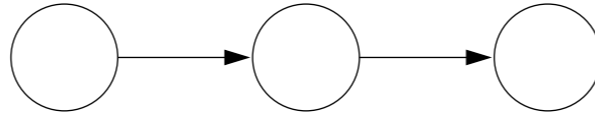
Applications:

- Games: Chess, Go, Pac-Man, Starcraft, etc.
- Robotics: motion planning

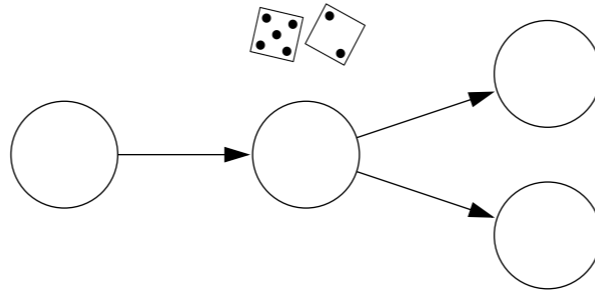
- Reflex-based models are too simple for tasks that require more forethought (e.g., in playing chess or planning a big trip). State-based models overcome this limitation.
- The key idea is, at a high-level, to model the **state** of a world and transitions between states which are triggered by actions. Concretely, one can think of states as nodes in a graph and transitions as edges. This reduction is useful because we understand graphs well and have a lot of efficient algorithms for operating on graphs.

State-based models

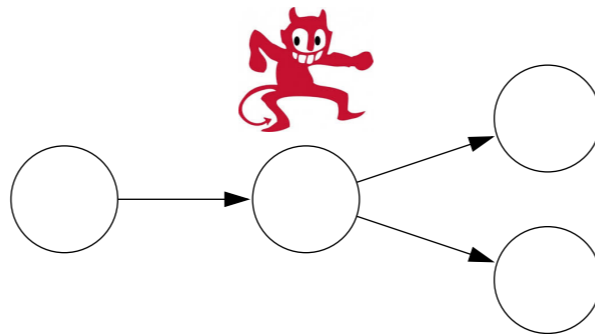
Search problems: you control everything



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



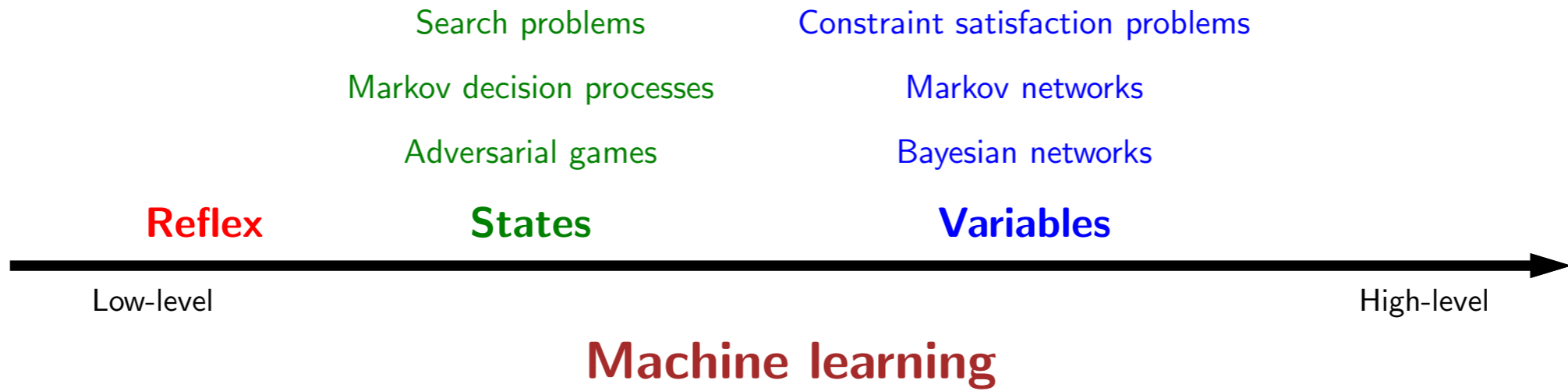
Adversarial games: against opponent (e.g., chess)



- We consider three types of state-based models.
- **Search problems** are models where you are operating in an environment that has no uncertainty. However, in many realistic settings, there are other forces at play.
- **Markov decision processes** handle situations where there is randomness (e.g., Blackjack).
- **Adversarial games**, as the name suggests, handle tasks where there is an opponent who is working against you (e.g., chess).

- In one of the homeworks, you will build an agent that plays Pac-Man.

Course plan



- 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

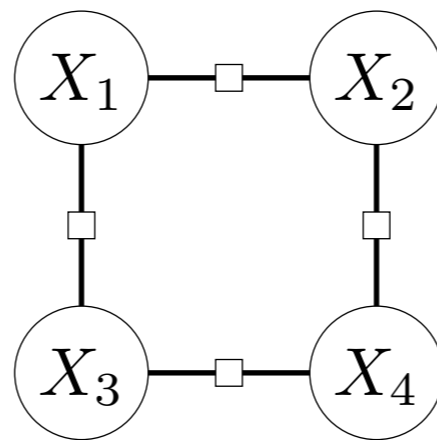
Goal: put digits in blank squares so each row, column, and 3x3 sub-block has digits 1–9

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

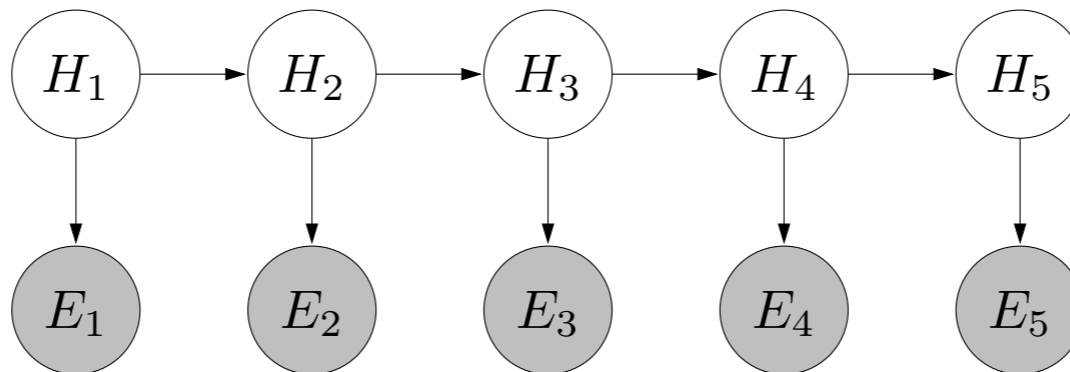
- In state-based models, solutions are procedural: they specify step by step instructions on how to go from A to B.
- In some applications, the order in which things are done isn't important.
- For example, in Sudoku, where the goal is to put digits in the blank squares to satisfy some constraints, all that matters is the final configuration of numbers; you can fill them in in any order.
- Variable-based models allow you to declare you want (it's like a higher-level language) rather than micro-manage how you want the solution to be found.

Variable-based models

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

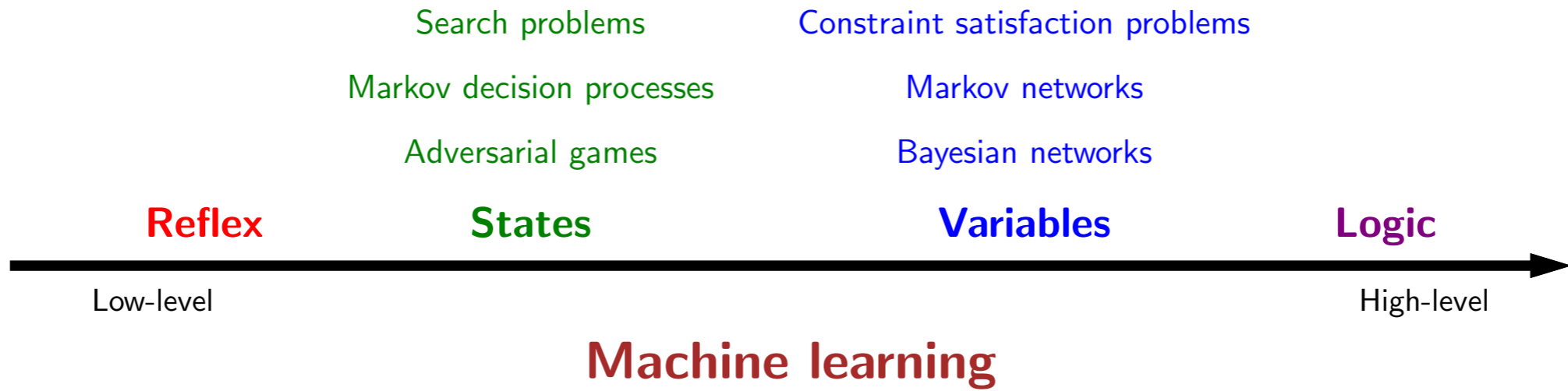


Bayesian networks: soft dependencies (e.g., tracking cars from sensors)



- **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 variables are random variables which are dependent on each other. For example, the true location of an airplane H_t and its radar reading E_t are related, as are the location H_t and the location at the last time step H_{t-1} . The exact dependency structure is given by the graph and it formally defines a joint probability distribution over all of the variables.

Course plan



- The last topic is logic.

Motivation: virtual assistant

Tell information



Ask questions



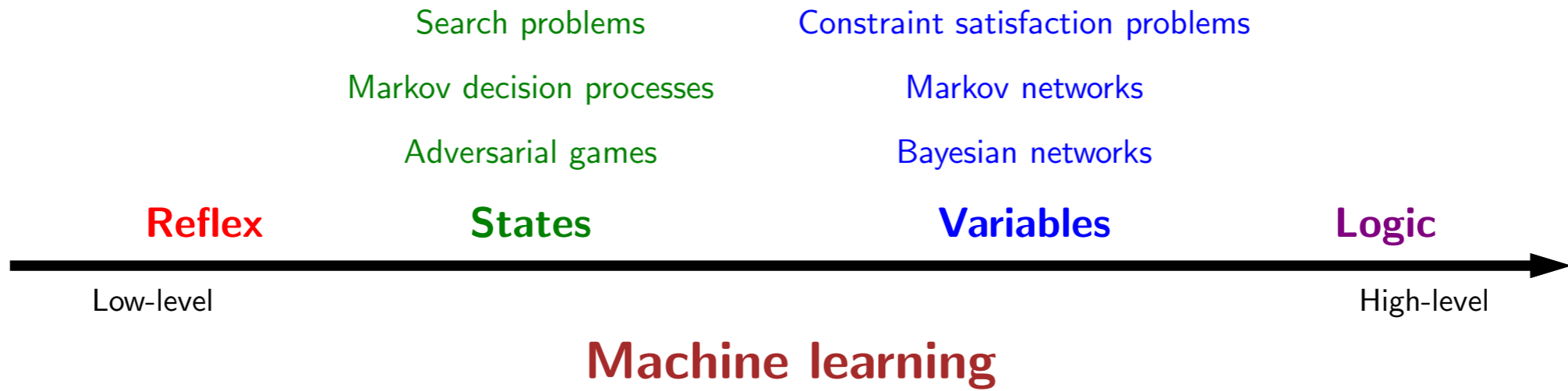
Use natural language!

Need to:

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

- One motivation for logic is a virtual assistant. A good assistant should be able to remember what you told it and answer questions that require drawing inferences from its knowledge. And you'd want to interact with it using natural language, the tool that humans invented for communication.
- Interacting with this system feels very different than a typical machine learning-based system. First, it is adaptive, whereas most ML systems are a fixed function. Second, the types of information we provide it and the types of questions are more heterogeneous and more abstract, and we expect the system to realize the full consequences of every single word. (We don't want to tell our personal assistant 100 times that we prefer morning meetings.)
- One often contrasts logical AI and statistical AI. In this course, we will treat the two as not contradictory but rather complementary. Logic provides a class of models which is higher-level, but still needs to be supported by the groundedness to real data that machine learning offers.

Course plan



- And this concludes our tour of the topics.
- To summarize, we will discuss models, going from reflex to state-based models to variable-based models to logic. For each, we will instantiate the modeling-inference-learning paradigm.



Overall Summary

- Course Logistics
- History: roots from logic, neuroscience, statistics—melting pot!
- AI has high societal impact, how to steer it positively?
- Modeling [reflex, states, variables, logic] + inference + learning paradigm