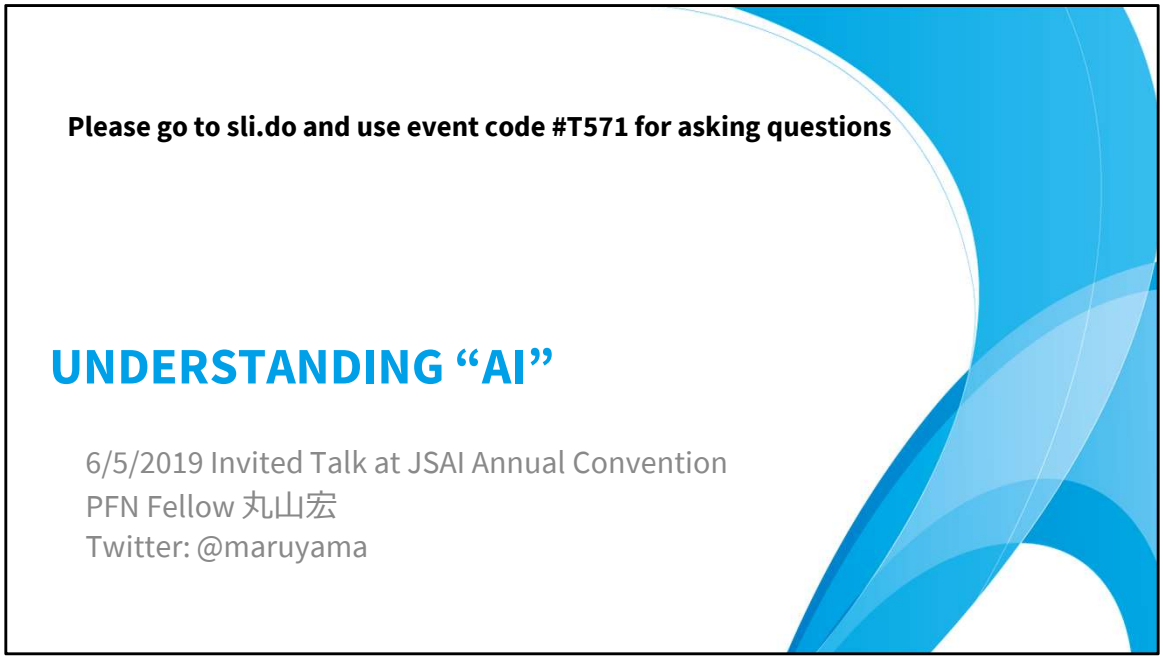



How to understand “Artificial Intelligence?”

## Agenda

1. What is “AI”
2. New Computation Model
3. Implications to Science and Engineering

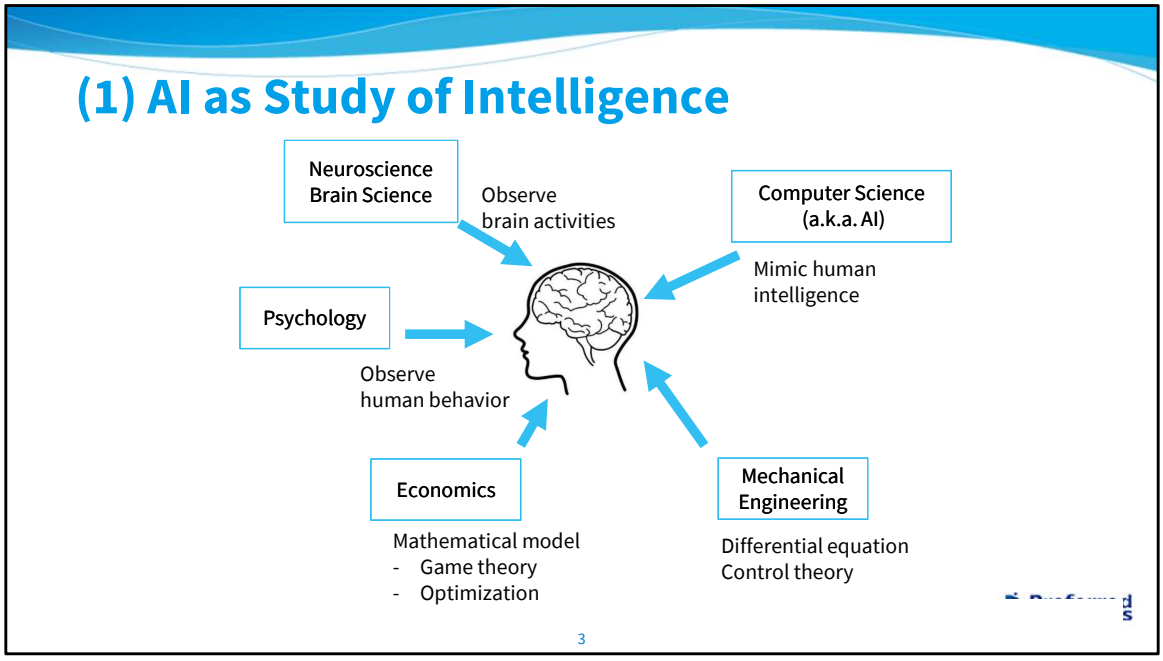
Preferred Networks



In this talk, first I point out that the word “AI” is used in many ways and it is causing confusion.

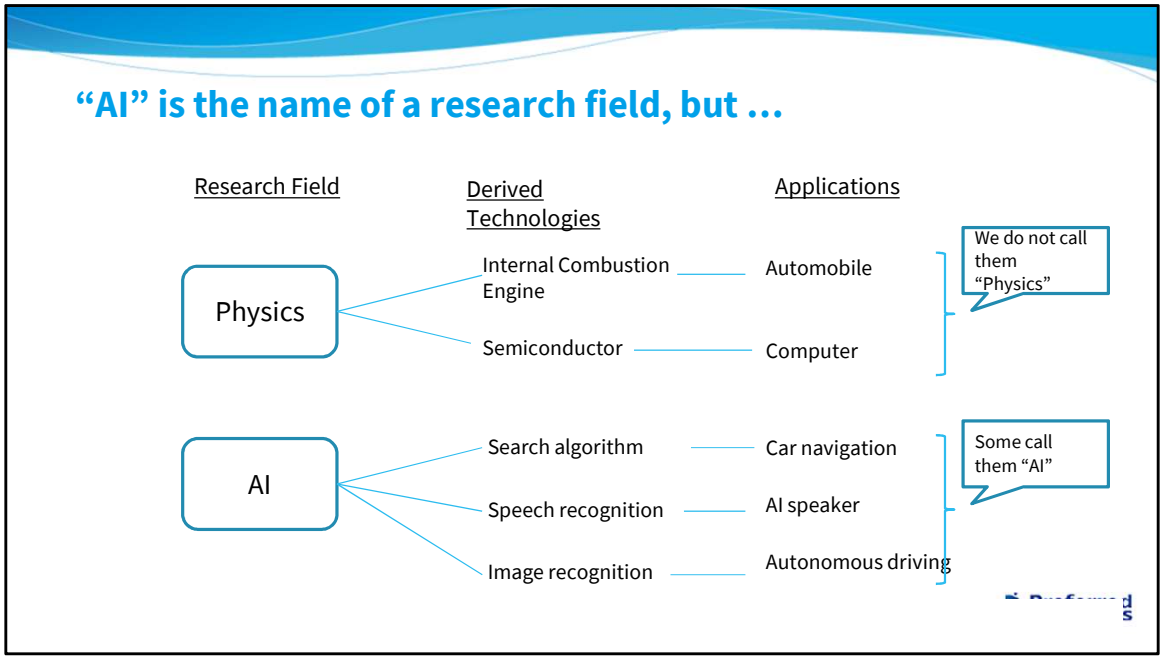
Then, I will focus on “AI as frontier of computer science” and argue that a new computation model is emerging.

Finally, I will discuss how this new computation model affects on science and engineering.



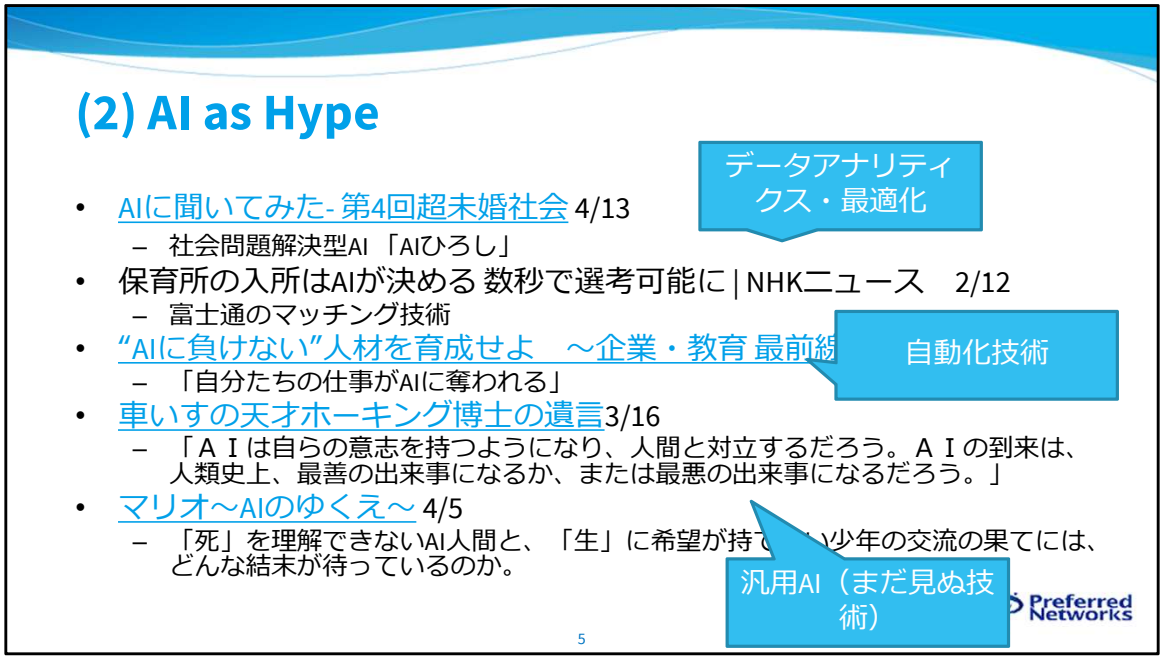

The term “AI,” or Artificial Intelligence, is used in different meanings in different contexts.
There are at least three different interpretations.

First, AI is a research field to study “intelligence.”
There are several different approaches to intelligence; neuroscience tries to understand intelligence by observing neuron activities in a brain, psychology by observing human behaviors, economics models human intelligence as an independent and rational agent, robotic researchers use differential equations and control theory to understand human motions, and we, computer scientists try to understand intelligence by building a software mimicking human intelligence.




AI is a research field just like physics, but there is one important difference. In physics, there are technologies derived from physical theories, and there are products based on these derived technologies, but they are not called “physics.”

On the other hand, in AI, the derived technologies as well as products are sometimes called “AI,” and THAT is the source of confusion.

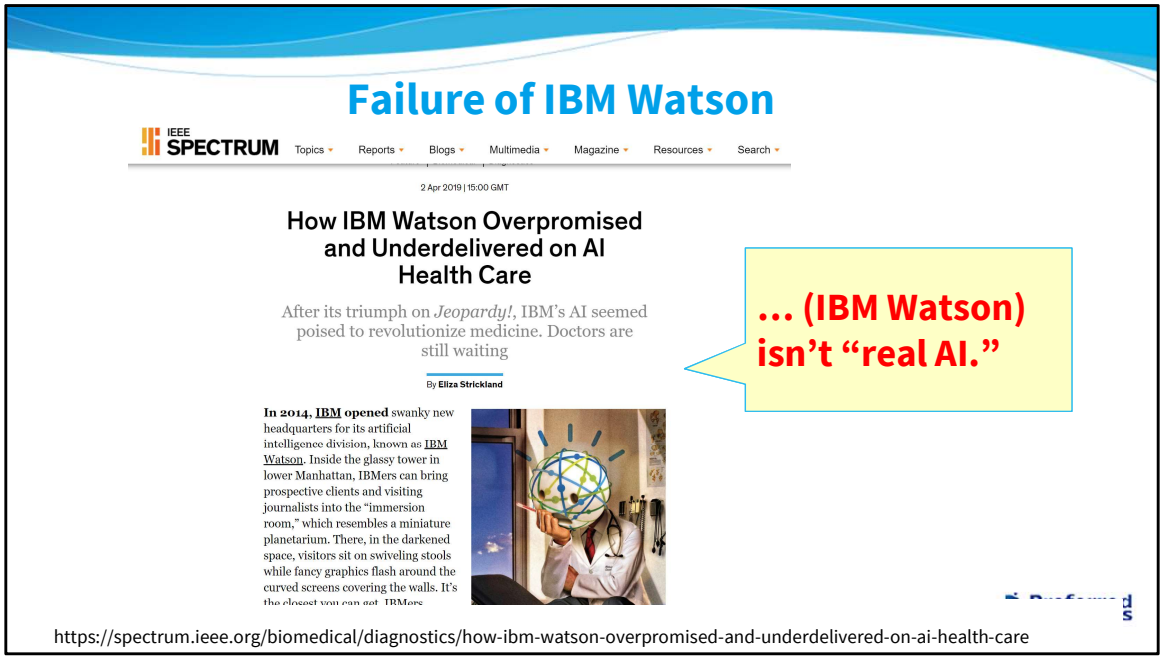



That leads to the second interpretation – "AI as something sophisticated." Here are some examples of recent coverage by NHK.

In the program "Ask AI," statistical analysis of social data is used for drawing some recommendations for social issues.
The news on "Nursery Matching AI" refers to a combinatorial optimization problem.
Discussion on how job security is threatened by AI is actually talking about automation.
The program on Dr. Hawking's Last Will and the drama "Mario the AI" are dealing with hypothetical technologies that have never been realized.

The reason I call this interpretation "AI as a hype" is that, in this use, people are less concerned with what "AI" is. The speaker means one thing, and the listener may understand it as completely another, and they do not bother.

If the speaker uses "AI" in order to intentionally confuse the listener, tragedy happens.




This is an IEEE Spectrum article published in April 2019.
It reports on the spectacular failure of IBM Watson for Healthcare.
One customer, University of Texas MD Anderson Cancer Center, had spent 62 million dollars on a Watson project before it was finally canceled.

One notable comment of the customer is "(IBM Watson) isn't a real AI" – whatever they mean by "real AI."
In fact, IBM Watson is a very good technology – as an information retrieval system.
But I presume that IBM's sales team sold it as an "AI," while they were fully aware that the word would give the customer an impression that it could do more than technically feasible.


https://www.youtube.com/watch?v=gapHaWwtGnM, accessed on 2019/5/15

Preferred Networks

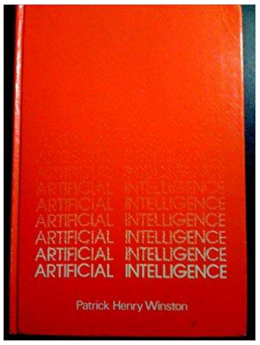
7

There is even a company who call their technology “Artificial General Intelligence,” or AGI (at least in Japanese translation).

I think it is the responsibility of everybody in this room to communicate the right message about our technology. We should not oversell or undersell our technologies. If the sales team or the PR department of your company is sending a wrong message to their customers or the society, it is YOUR responsibility to correct it.

Yesterday, Dr. Uramoto, the president of JSAI, gave a presentation on ethics of AI researchers.
I think sending right messages should be at the top in our priority list of ethics.

## (3) AI as Frontier of Computer Science




P. H. Winston, *Artificial Intelligence*
**ISBN-13:** 978-0201084542, 1977

Frontier in 1970's

Source: Wikipedia

Recursive call was AI Technology

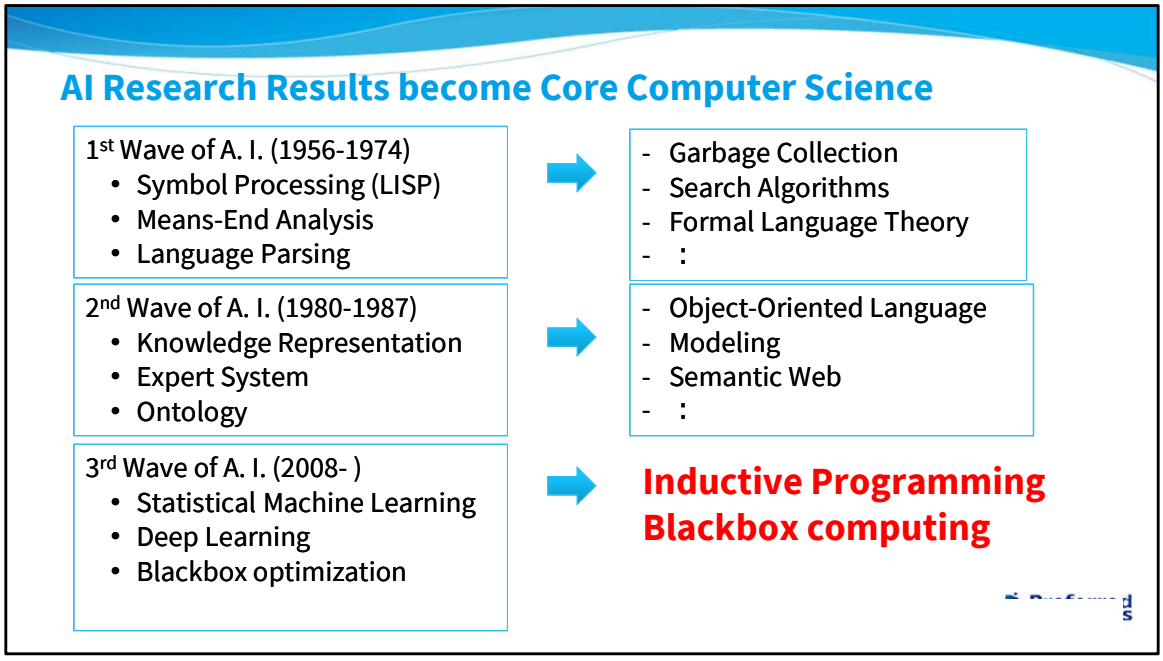
The third interpretation of "AI" is "AI as frontier of computer science (CS)."

Many CS ideas were came from AI research.

This is one of the early textbooks on AI.
It covers things like

- Symbolic programming (LISP programming language and dynamic memory management i.e., garbage collection)
- Recursive call (e.g., to solve the puzzle "Tower of Hanoi")
- Search algorithm (e.g., depth-first search, A* algorithm, alpha-beta pruning, means-end analysis)
- Syntax analysis

All of these technologies are now in the core computer science and people do not call them "AI" anymore.




There have been three waves of AI research.

In the first wave of AI, research topics include
- Symbolic programming (lisp)
- Means-end analysis
- Natural language processing

and they are incorporated into the core CS as
- Dynamic memory management (garbage collection)
- Search algorithms
- Formal language theory

In the second wave of AI, research topics include
- Knowledge representation (e.g., Frame theory)
- Expert system
- Ontology

And they are incorporated into the core CS as
- Object-oriented programming
- Modeling language
- Semantic Web

So, if you look at what is happening in AI research, you can predict where CS is to go.

In the current wave of AI research, focus areas include
- Statistical machine learning, especially deep learning
- Blackbox optimization

And I believe that these technologies will lead us to a new programming paradigm, or new computation model.

## Agenda

1. What is “AI”
2. New Computation Paradigm
   - Deep Learning
   - Blackbox Optimization
3. Implications to Science and Engineering

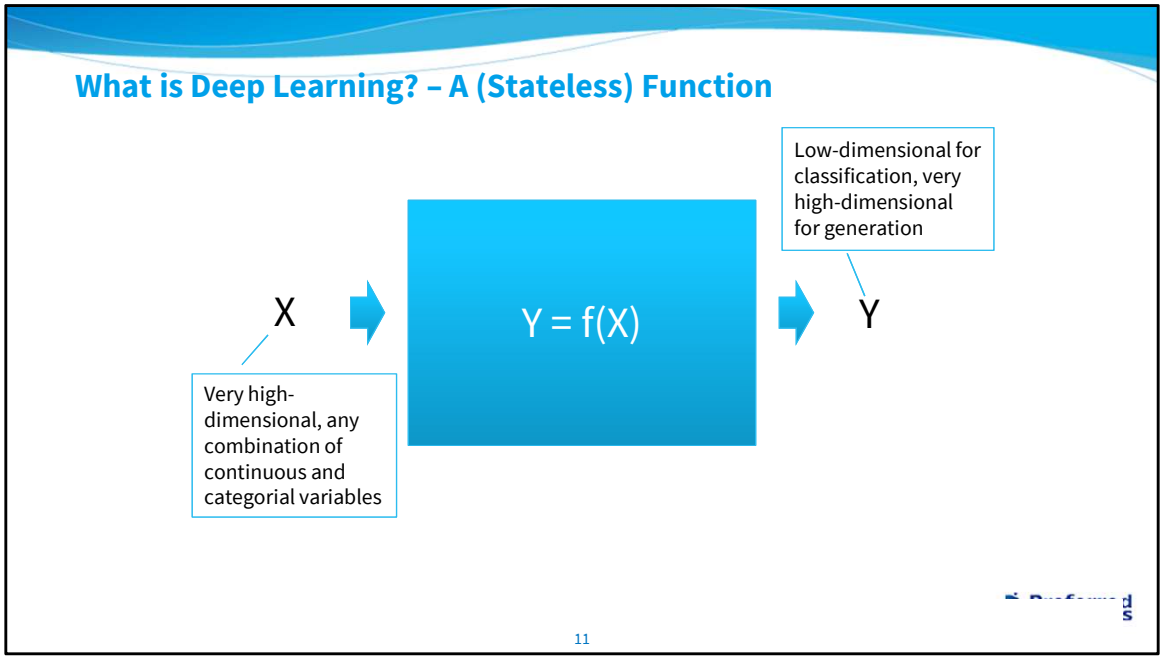
10

Preferred Networks

I said AI is to advance computer science.

Let me discuss these two technologies that are currently driving CS.
- Deep learning
- Blackbox optimization

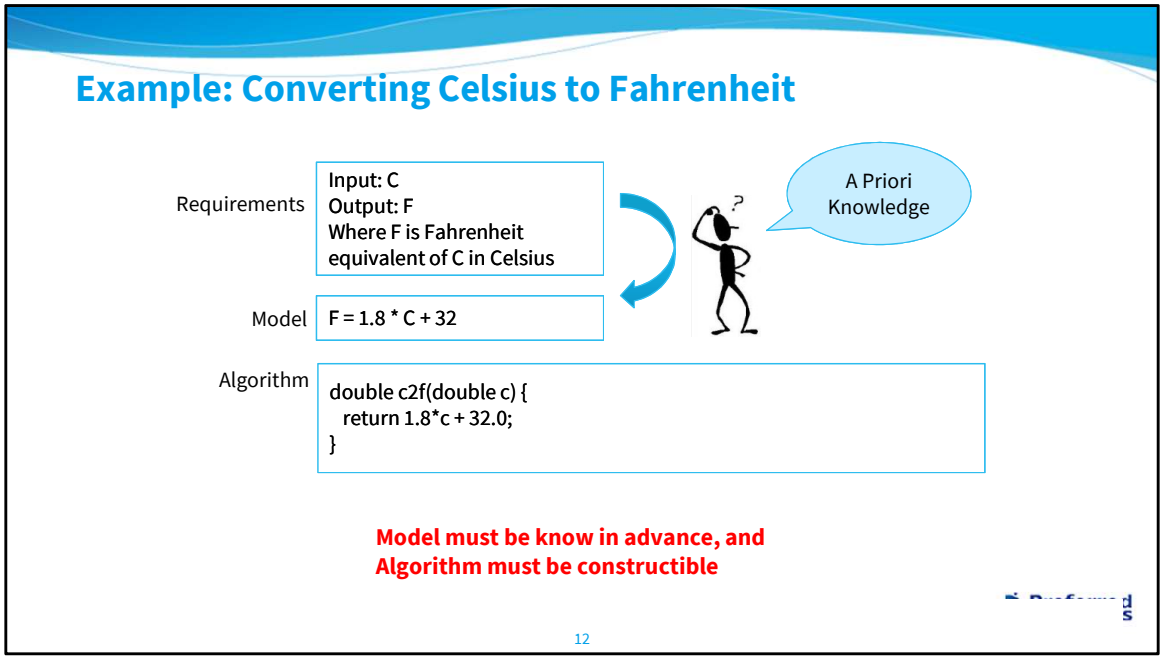
and how they will make a new computation model.




First, deep learning.

Very simply put, deep learning is a function.
A function takes X as input and produces Y as output.
Usually we are dealing with a very high dimensional vector as input. For example, if the input is a image of 100x100 pixels, the input is a vector with 10,000 variables.

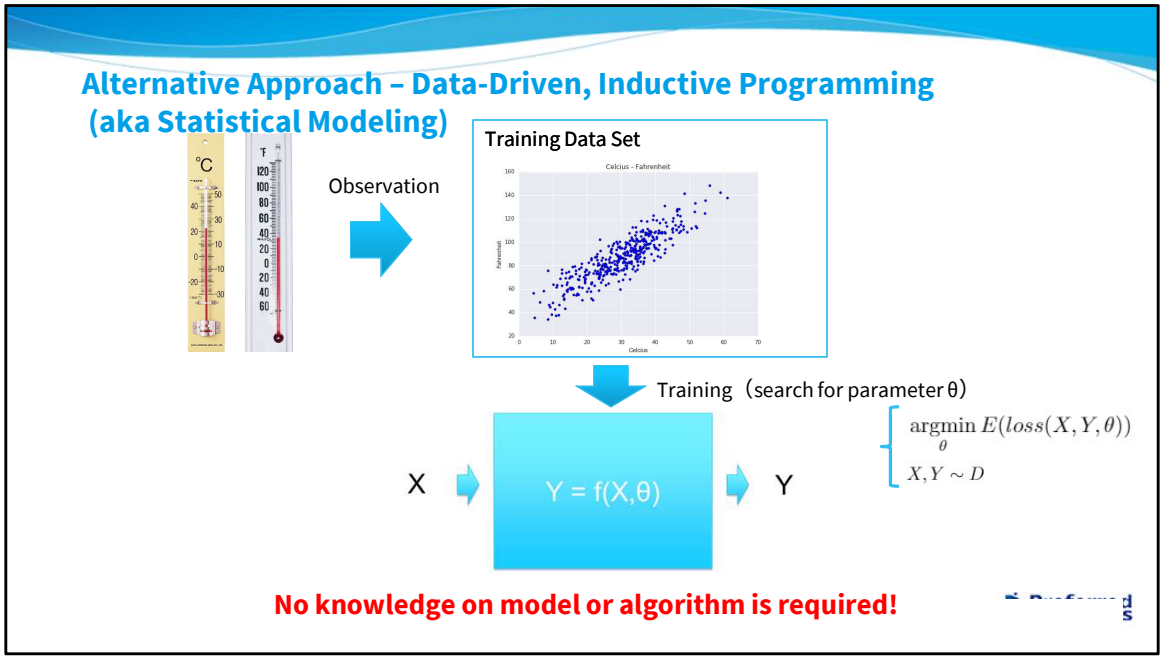
Output Y could be simple – if the function is to classify the input image into either a cat or an airplane, the dimension of Y is 2. If the output is also an image, the dimension is much higher.




Let's consider programming a very simple function – computing Fahrenheit from Celsius.

In conventional programming, which I call here "deductive programming," we start from the requirement – what is input and what is the output.
Then, we convert it into a "model" – by which I mean precise mathematical relationship between the input and output.
In this case, we use our knowledge on the relationship between Celsius and Fahrenheit and represent it as a mathematical formula.
Once you have the model, you can build an algorithm to actually compute it.

Note that in deductive programming, we assume that the model is known and the algorithm is constructible.




On the other hand, with deep learning, programming is done by giving examples – which I call inductive programming..

In our case, first we obtain two thermometers, one in Celsius and the other in Fahrenheit, and read their measurements from time to time.
Then you get input-output examples, which we call a training data set.

Once the training data set is ready, we train a deep neural net semi-automatically.

Note that inductive programming is possible even when we do not know the model or the algorithm. This opens opportunities to solve very interesting real-world problems.

## Model is unknown


自動運転のためのセグメンテーション
https://www.youtube.com/watch?v=lGOjchGdVQs

Preferred Networks

Here are some examples of real world applications of inductive programming.

First is semantic segmentation for autonomous driving. The top left panel is the video input. For each pixel of the input, our function has to calculate what class, such as road surface (purple), sidewalk (pink), vehicle (blue), tree (green), etc., the pixel belongs to. The output is shown in the right bottom panel.

In this task, it is very hard to define a proper model, because you can not give precise mathematical conditions with which a particular pixel is classified as “road surface.”

Instead of a model, we supply examples. The bottom left panel shows a human-annotated image, meaning that human annotators gave the correct answer to each pixel. We should supply thousands of these human-annotated images to get high accuracy. Even though we do not know the model, we can prepare training data set and obtain reasonably accurate results.


The second example is a voice-controlled robot. In this demo, the operator says “Can you move the brown fluffy thing to the bottom?” to move the teddy bear. Note that the top and bottom are from the operators perspective, as shown in the top view image of the boxes as shown on the left.

Next, she says “Can you move the tissue box to the left?” and then use another expression “white and blue box” to refer to the same object.

Again, this system is very hard to program in the conventional way, because we cannot encode all the possible natural language expressions in the system.

How did we train this system? We show the top view of the boxes and ask human annotators “what would you say if you want to move this to there?” and collected thousands of example sentences. With these example sentences as a training data set, we trained the system.

## Algorithm is Unknown

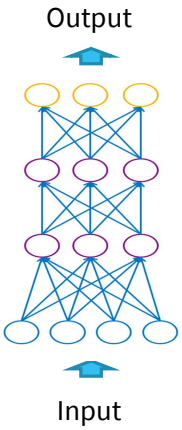

The third example is where the model is known but the algorithm is unknown.
This is an application to colorize a line drawing. The left image is a line-drawing input and the right image is the output. You can also give hints in terms of what color is used in each region.

How did we train this neural net? We collected a lot of Anime images from the Internet and applied a known algorithm called “edge detection” to obtain the corresponding line drawing. Using many such pairs we trained our neural net.

Because the edge detection algorithm is well known, the mathematical relationship between a line drawing and the corresponding color image is known in a sense -- that is, the model is known.

However, how to compute a colored image from a line-drawing input is not known, meaning that the algorithm is unknown.

## Deep Neural Net as a Universal Computing Mechanism




- Very large number of parameters
- Can approximate ANY high-dimensional function*

➔ **Pseudo Turing Complete!**

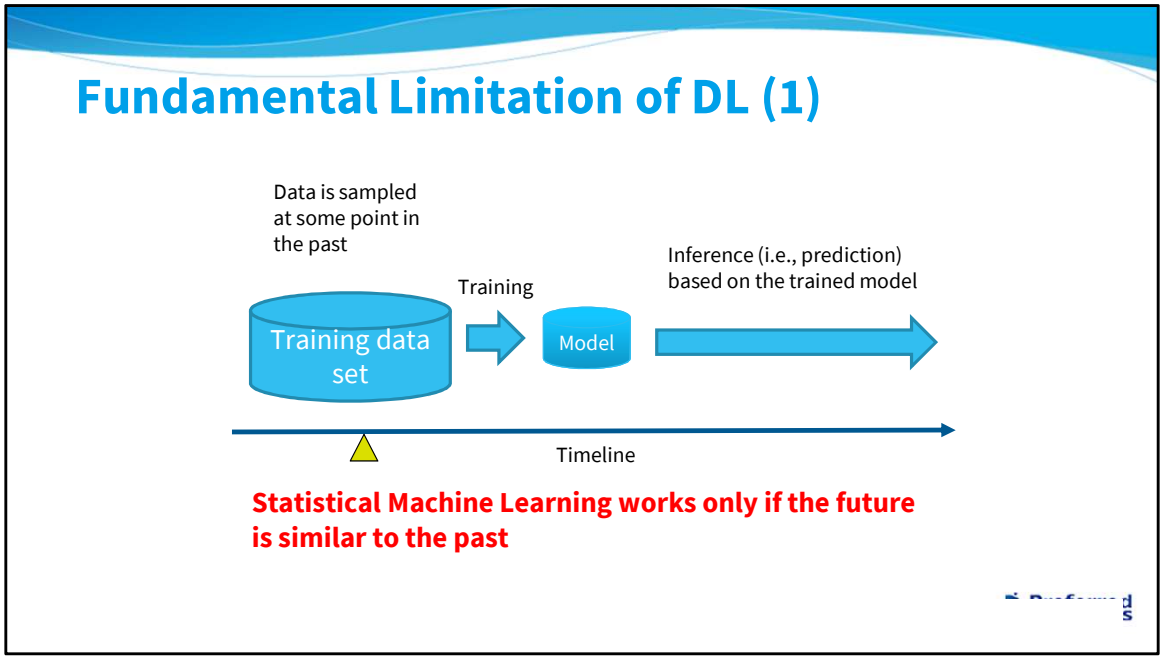
* G. Cybenko. Approximations by superpositions of sigmoidal functions. Mathematics of Control, Signals, and Systems, 2(4):303–314, 1989.

17

I said that deep neural net is a function. How powerful is it? Is it powerful enough to represent any computable functions? The answer to this question is yes.

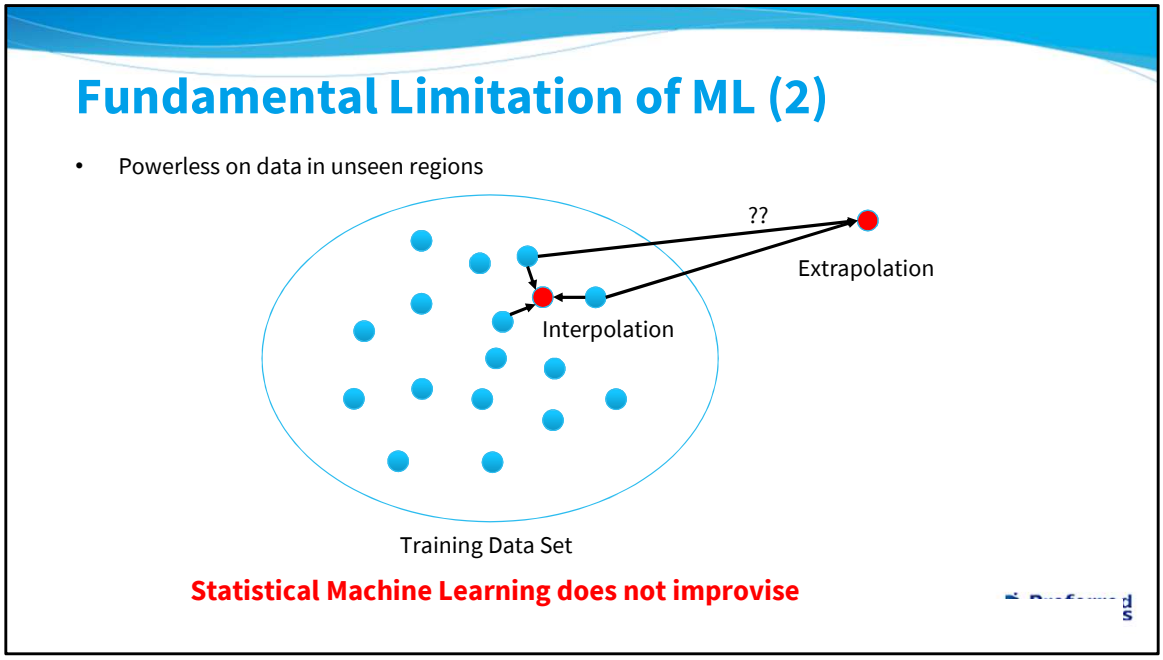
Because a deep neural net has a very large number of parameters, it is theoretically proven that for any computable function, there exists a complex enough deep neural network that approximates the function within the given error margin.

In other words, deep neural network is “pseudo Turing complete,” meaning that we have a universal computational mechanism, which can be programmed with examples. This is a significant breakthrough in computation, I believe.




Even though deep learning is a very powerful computing mechanism, it is not an almighty intelligence. It is statistical modeling and hence, there are fundamental limitations. I will explain three such limitations.

First is that deep neural net is trained using the data observed in the past and use the trained deep neural net (DNN) to predict future events. Thus, we always assume that the future is essentially the same as the past.




The second limitation is this.

Even if the future and the past look the same, the training data set is always finite. If we are dealing with an infinite set as the domain of the function, which is almost always the case, an input to the function is in general something that is never seen in the training data set. If the input is close to some of the data points in the training data set, the prediction is very accurate. This situation is called “interpolation.”

On the other hand, if the input is far from the any of the data points in the training data set, the prediction is very hard. This situation is called “extrapolation.” So deep learning is powerful for frequent events but powerless for rare events that do not appear in the training data set.

Often customers ask us to build an “AI” system that can deal with emergency situations. As long as deep learning is concerned, it is impossible.  We always need a lot of “emergency” samples in the training data set.

## Fundamental Limitation of ML (3)

- Always works statistically


**No guarantee of "100% correctness"**

20

The third and probably the most serious limitation is that the computation of deep learning is essentially probabilistic. Because we consider deep learning as statistical modeling, we always assume that there is an original probability distribution which is unknown. We further assume that the training data set is created by random sampling, or i.i.d. (Independent and Identically Distributed). After we obtained the training data set, the DNN is trained with a known training algorithm.

But the problem is in the first "random sampling" step. This is random sampling, so there is always bias. For example, when you throw a dice for a hundred times, what is the chance of getting 'one' for all the hundred throws? It is very unlikely, and you may say that it will never happen. But there is no guarantee that it never happens because of the bias. This means that deep learning has no guarantee of its correctness. There is no such thing as 100% correct deep neural net.

## What is Deep Learning – Recap

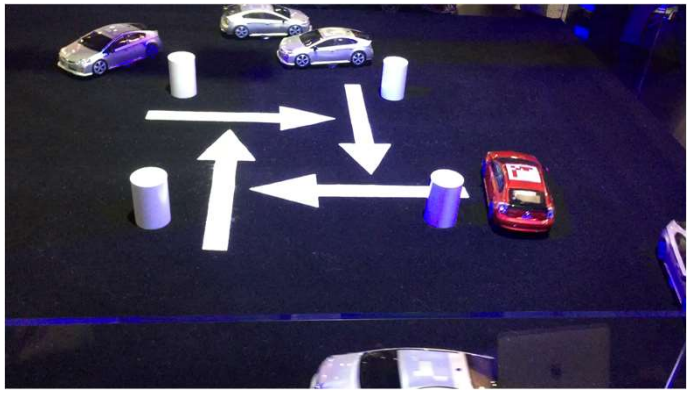
- A new way of programming (*inductive* programming)
  - No prior knowledge on model or algorithm
  - Creative “teacher signal” allows innovative applications
- It’s statistical modeling
  - Assume i. i. d. (independent and identically-distributed)
  - Approximation only (no exact answers)

21

Here is the recap of what deep learning is.
Deep learning is a new programming model. It is inductive, not deductive, and the programming can be done without knowing the model or the algorithm. Innovative thinking on how to give teacher signals may solve many interesting problems.

On the other hand, it is fundamentally a statistical modeling. It assumes that the original distribution is sampled i.i.d.
There is always bias, and what you get is always an approximation.

Blackbox Optimization: Reinforcement Learning for Autonomous Driving



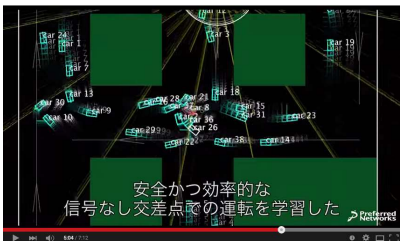
Consumer Electronics Show (CES) 2016

Let me turn to the second technology that current AI research focuses on: blackbox optimization.

This is an autonomous car demo at Consumer Electronics Show in 2016.
The silver cars were trained using deep reinforcement learning.
The red car was controlled manually by our vice president, and he sometimes hits or blocks other cars.
But as you can see the silver cars can avoid collisions.

## External Simulator as an Oracle

S: Current State




u(S, Y): Reward function

X: Sensor Input

Y: Actuator Output

$Y = f(X, \theta)$

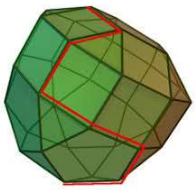
$$\begin{cases} \underset{\theta}{\operatorname{argmax}}\, E(u(X,\theta)) \\ X \sim D \end{cases}$$

23

How this system was trained? We used an external simulator. The current situation in the simulator is supplied as a simulated sensor input X to the learning algorithm. The learning algorithm compute the actuator output Y so that it maximize the reward function, or the utility function, u(S,Y). Thus, this is an optimization problem.

# Blackbox optimizers

Whitebox Optimization
- Simplex algorithm
- Internal point method



The utility function is known in advance

Blackbox Optimization
- Multi-arm bandit problem
- Genetic algorithm
- Reinforcement learning
- Bayesian optimization

x

u(x)

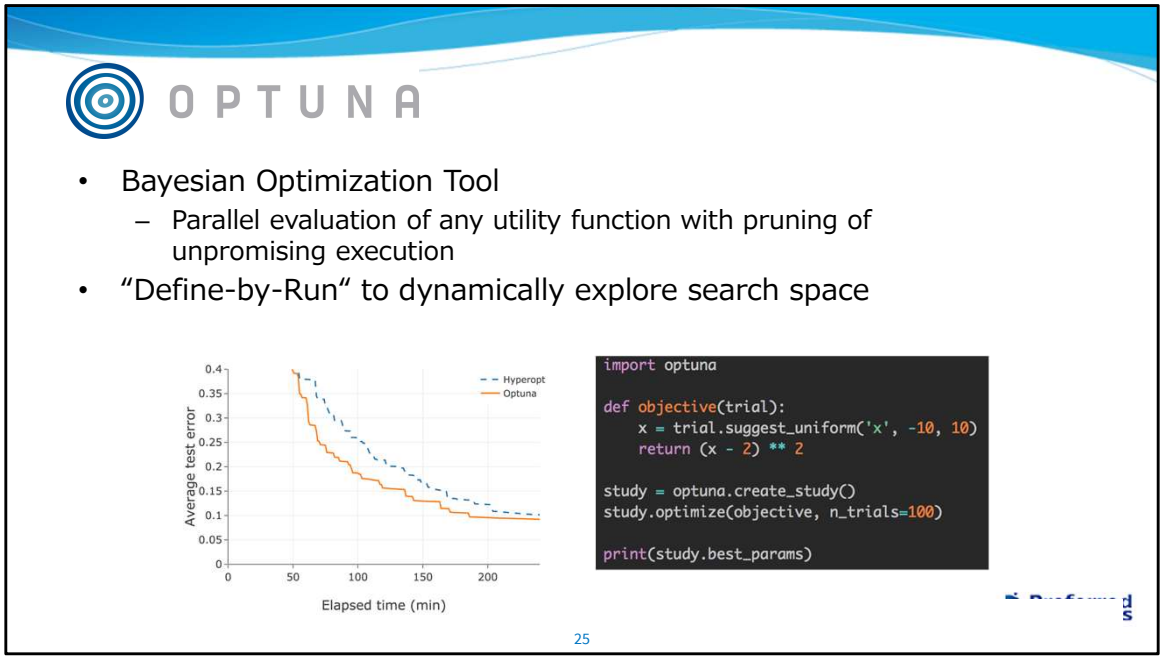
- Utility function is not known in advance
- Use an external oracle for individual utility values

24

This type of optimization problem is much different from the traditional optimization problems in operations research, such as linear programming (simplex method) and convex optimization (interior-point method) where the utility function is known in advance.

Blackbox optimization, such as reinforcement learning, assumes no prior knowledge on the internals of the utility function – thus it is called "blackbox." Instead, it receives signals on the utility function by asking an external "oracle" the value of utility function u(X) for a given parameter X.

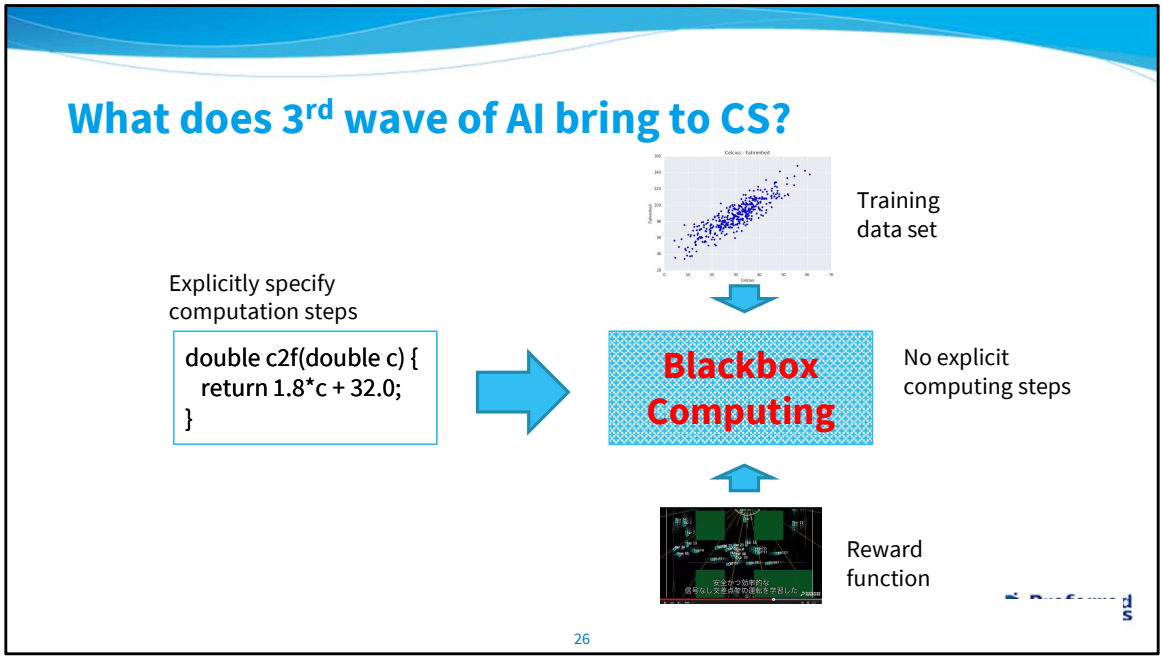
The idea of blackbox optimization is not new. Multi-arm bandit problem and genetic algorithm can both be considered as blackbox optimization. More recently, a new technique called Bayesian optimization attracts a lot of attention, especially in the context of hyper-parameter tuning in deep learning.




For example, Optuna is an open-source technology of Bayesian optimization. It allows very computationally-expensive oracles, such as deep learning, with pruning of unpromising trials.

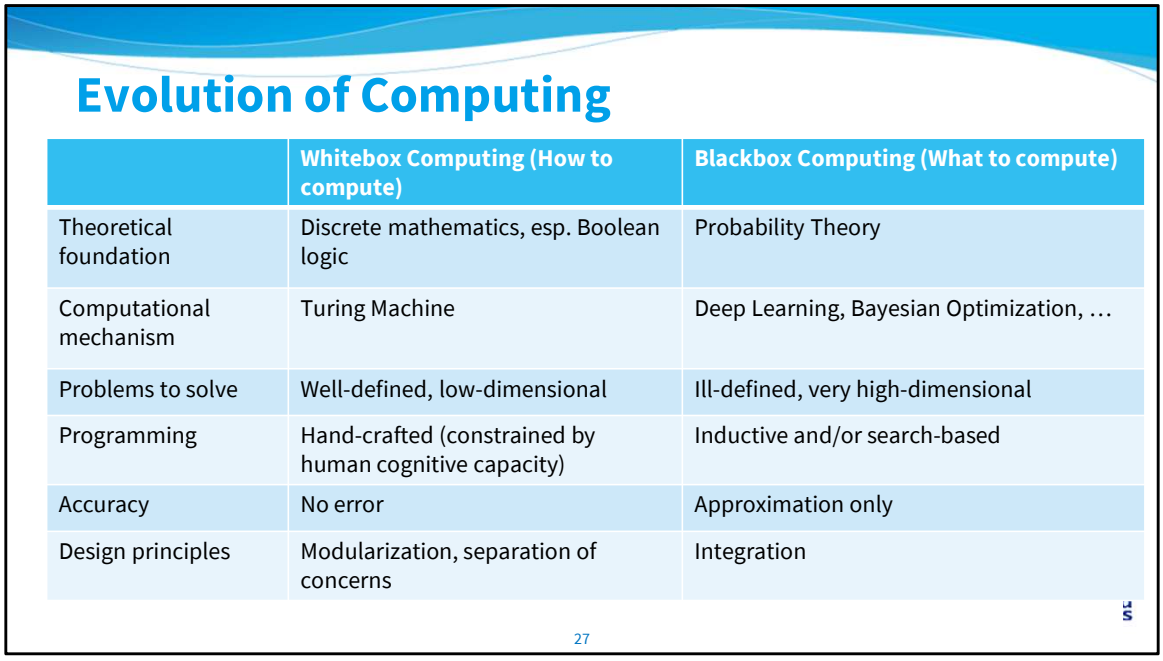
It also features “define-by-run,” meaning that the exploration space can be dynamically constructed as the exploration proceeds.

Blackbox optimization is a very powerful tool. It does not assume anything about the utility function, yet it can optimize for it.
In a sense, with blackbox optimization you specify what you want but not how to solve it. The “how” part is automatically discovered by the underlying algorithm.




With these two technologies arisen from AI research, namely deep learning and blackbox optimization, we are now witnessing an evolution of computing.

That is, from
what I call whitebox computation where you have to specify “how to compute,”
to
what I call blackbox computation where you only specify “what to compute” and let the machine to figure out how to compute it, which I call “Blackbox Computing.”




Revolution of computing – from whitebox computing to blackbox computing.

## Agenda

1. What is “AI”
2. New Computation Model
3. Implications to Science and Engineering

28

Preferred Networks

I discussed that a new computing paradigm, blackbox computation, is emerging.

If this is the case, then what are its implications to our society?
Here I discuss two areas -- science and engineering.

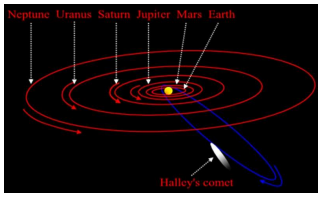
## cf. Evolution of Science

Law of Gravitation

$$F = G\frac{Mm}{r^2}$$

Occam's Razor

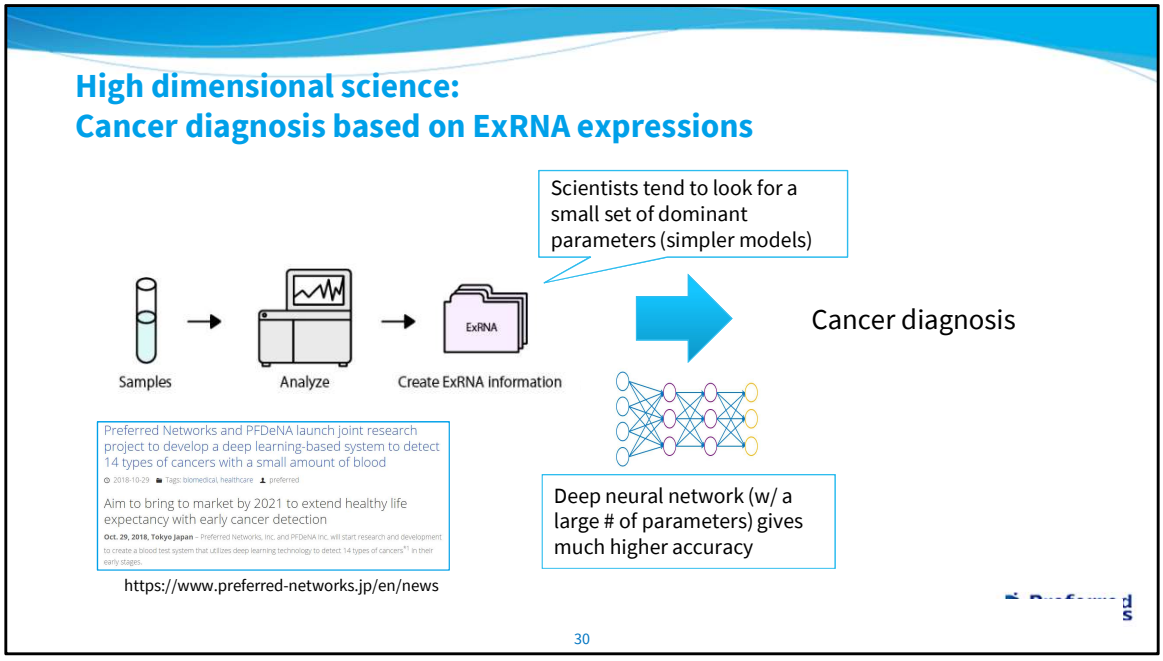
*"The explanation requiring the fewest assumptions is most likely to be correct."*




**Model with the smaller number of parameters is the correct one**

First, how scientific methodologies are affected by blackbox computation?

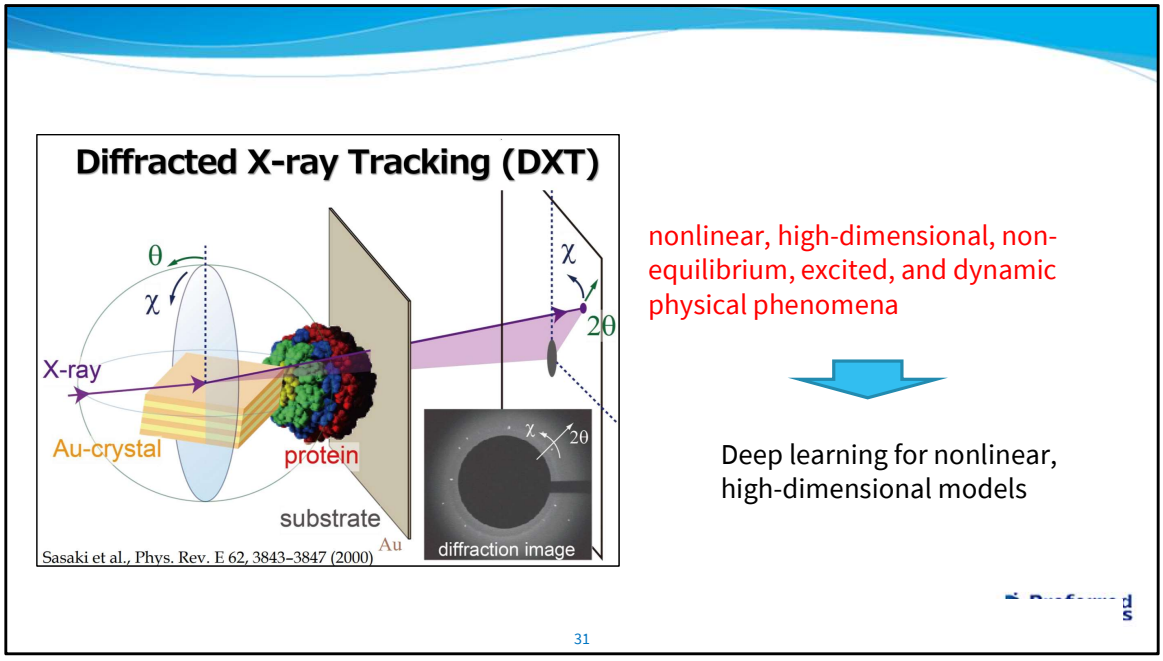
Take for example the law of gravity, Sir Isaac Newton's beautiful law. Why is this beautiful? Because this simple equation can explain both the movement of celestial bodies and a dropping apple. Behind this, there is a fundamental value in science called Occam's Razor which says "The explanation requiring the fewest assumptions is most likely to be correct."




On the other hand, many problems in the real world are not simple.

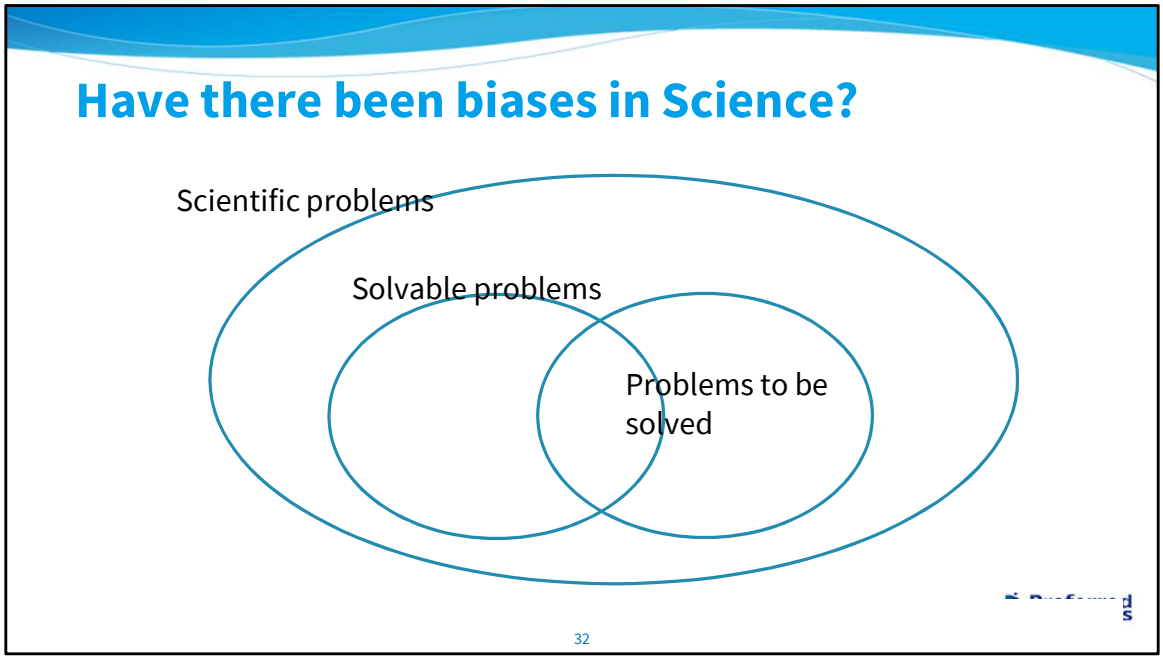
We are working with National Cancer Center to diagnose cancer by analyzing exRNA in blood.  Usually RNAs stay in cells but some fragments may move out of the cell. These RNA fragments are called exRNA. It is believed that by analyzing the expressions of these exRNA in blood we can diagnose various kinds of cancer.

There are more than 4,000 different types of exRNA, and scientists have been searching for a small number of predominant types that are indicative to a particular cancer.
Our team applied deep learning on this problem. We took all the 4,000+ types of exRNA into account and by looking at the whole pattern of these expressions, we could achieve order-of-magnitude higher accuracy of cancer diagnosis.

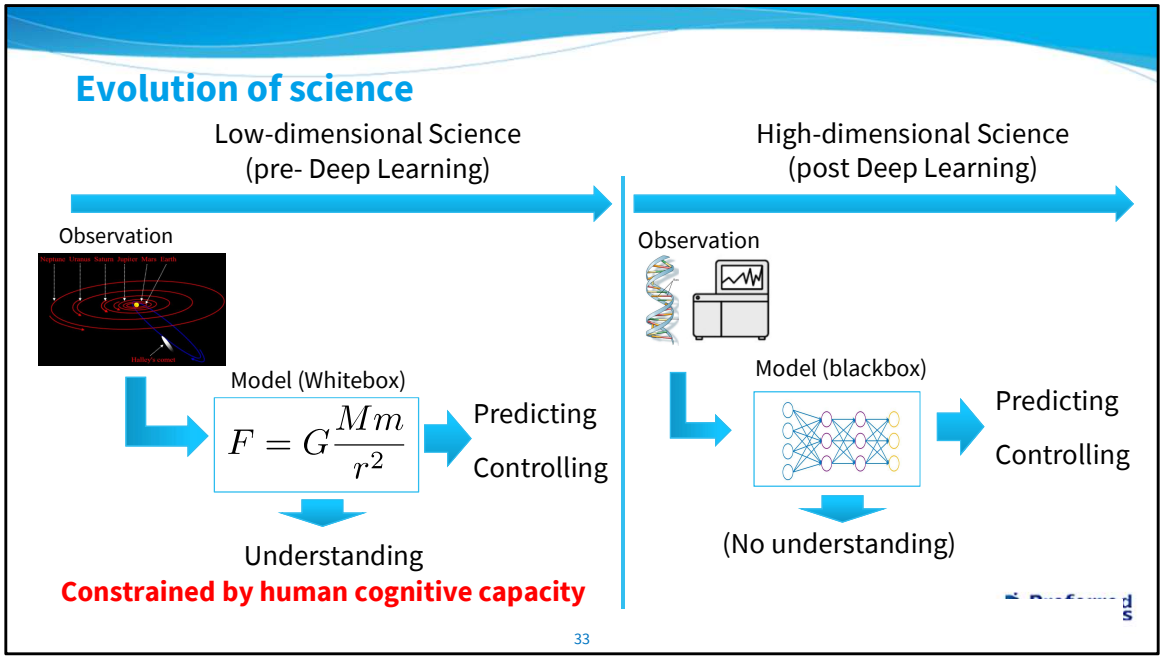



Some physicist told me that physical phenomena that is nonlinear, high-dimensional, non-equilibrium, excited, and dynamic is considered “dirty” and being avoided by many physicists.
Diffracted X-ray Tracking is one such problem – using very high-energy X-ray diffracted by tiny gold crystal attached to an endpoint of a protein molecule, you can observe the dynamic nature of the protein molecule. But this modeling has been very hard with traditional tools because of its nonlinear, high-dimensional, and dynamic nature.

With deep learning in our science toolbox, we have much better chance of analyzing this phenomena.




By looking at these complex problems, a question arises if our science has been focusing on problems that can be solved, not problems that should be solved.




What does it mean in terms of scientific methodologies?

In the past, we observed the mother nature and came up with a simple model on how the nature works. Once we understand the mechanism, we can use this knowledge to predict the movement of celestial bodies and control the trajectory of a cannon shell. The model should be simple enough to be understood by a human being. This means that in this mode of scientific activities, which I call low-dimensional science, the complexity of the model is limited by the human cognitive capability.

On the other hand, now we have a computational tool that can deal with high-dimensional models. I call this high-dimensional science. In high-dimensional science, we derive a model by applying, for example, deep learning, whose internal workings are too complex for a mere human to understand, but still useful for predicting or controlling the physical world.

Probably this is a controversial view – later today, there is a special session on AI and Physics, and I expect to see further discussions on this topic.


Last month I wrote a blog on high-dimensional science. If you are interested in, please read the blog (unfortunately it is in Japanese).

## The world is changing – "Software 2.0"

"Software is eating the world,"
Marc Andreessen, 2011

*Deep Learning is Eating Software*

November 13, 2017
By Pete Warden
Uncategorized
19 Comments

*Photo by John Watson*

When I had a drink with Andrej Karpathy a couple of weeks ago, we got to talking about

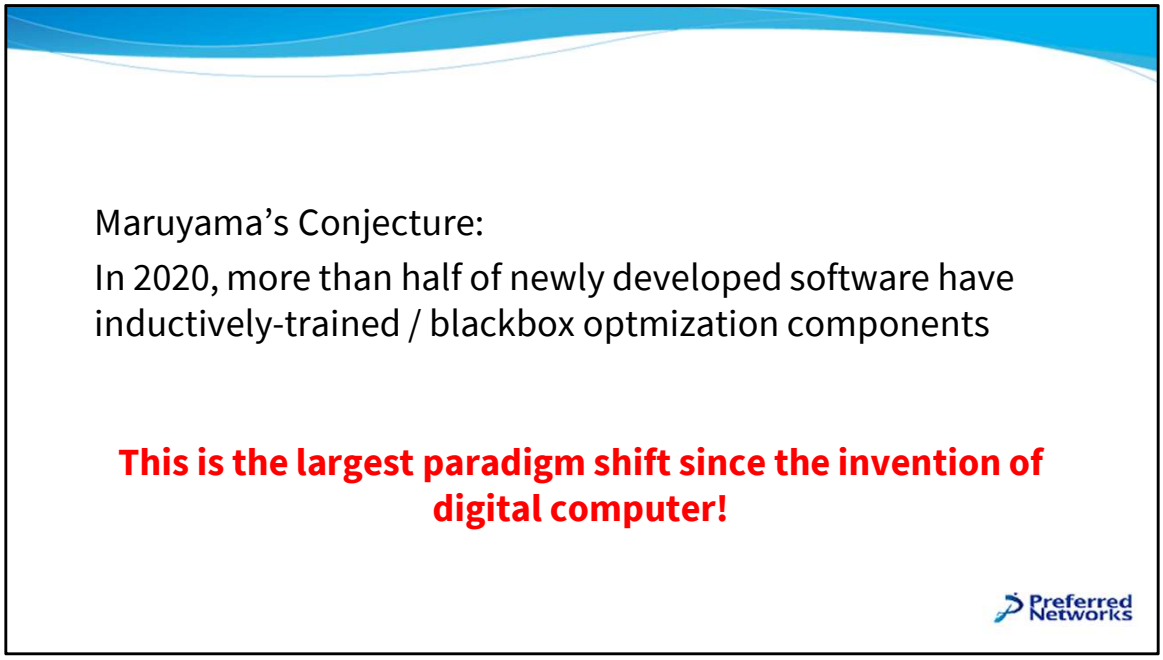
https://petewarden.com/2017/11/13/deep-learning-is-eating-software/

35

The second implication of blackbox computation is on the engineering side.

In 2011, Marc Andreessen said that "software is eating the world."
More recently Pete Warden said that "deep learning is eating software."

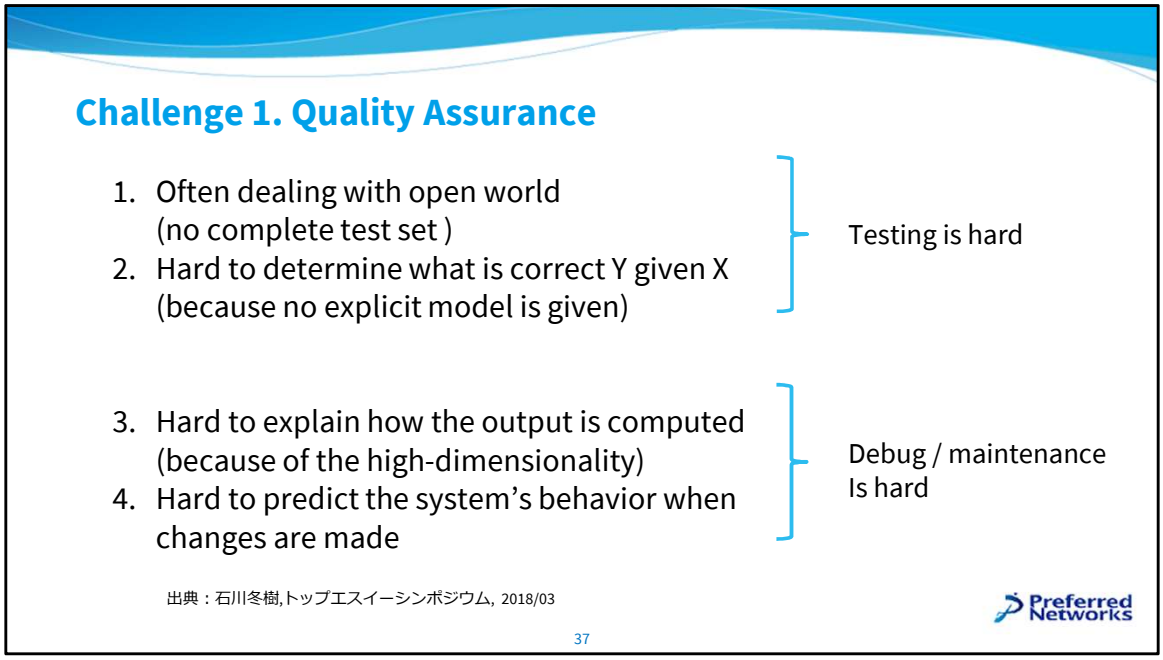
I think they are right.




This is my personal conjecture – by the end of the next 2020, more than half of newly developed software use blackbox computation in one way or another.

And this is the largest paradigm shift since digital computers were first invented 70 years ago.

The question is, how we can make good use of this new technology? There are many challenges on how to apply blackbox computing in real world applications.




One challenge is quality assurance.

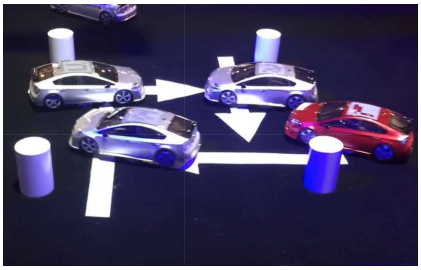
This is a slide taken from Prof. Fuyuki Ishikawa's presentation.
Because inductive programming often deals with real world problems and also it is inductive,

1. There are infinite number of requirements
2. We cannot have test oracles that are absolute
3. It is hard to explain how the output is computed
4. The behavior of the system is hard to predict.

These points make deep learning-based systems harder to test and debug.

## Challenge 2. Requirement Development

**What happens if we increase the collision penalty to the infinity?**



Cars that do not move!

**You have to be explicit in stating the balance between the utility and the safety**

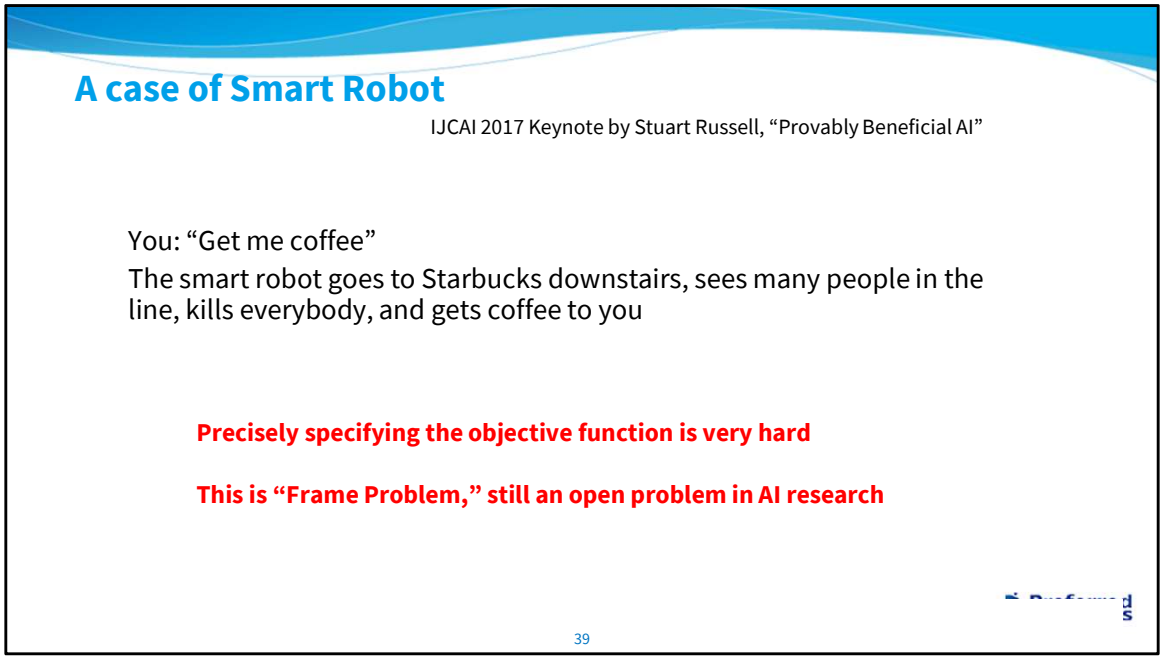
38

Another challenge is how to define the requirements for a blackbox optimization problem.

I showed you autonomous driving demo using reinforcement learning.
As I discussed, this system does not guarantee 100% avoidance of collision, because ultimately we use deep learning, which works statistically.

Then what happens if we increase the penalty of collision to infinity in order to make the cars “100% safe”?
The result would be cars that never move.
This is logical – if cars move, there are non-zero probabilities of collision. If you want make the probabilities down to absolute zero, the cars must not move.

This example reminds us that the safety and the utility are always tradeoff. In traditional system development, we pretend as if we can achieve both objectives simultaneously. If you formalize the task as an optimization problem, suddenly you have to explicitly quantify how much of safety can be sacrificed in order to achieve utility.



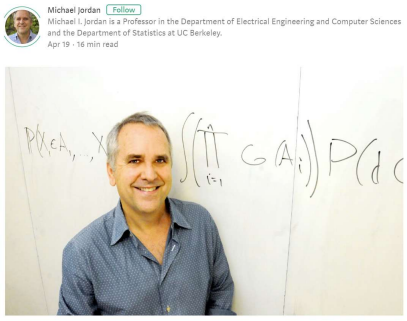

Another example is a fictious robot case, that was presented by Stuart Russell at IJCAI 2017.

Suppose there is a smart robot.
You say to the robot "get me coffee."
The robot goes downstairs to Starbucks, sees many people in the line, kills everybody, and gets coffee for you.

Of course you do not want the robot to kill people.
Instead of saying "get me coffee," you may want to say "get me coffee without killing people."
But there are infinite number of things that you do not want the robot to do.
This is actually one form of the "frame problem," an open problem in AI.

Both the autonomous driving case and the robot case tell us the difficulty of specifying the utility function in optimization.

## The need for new engineering discipline



Michael Jordan Follow
Michael I. Jordan is a Professor in the Department of Electrical Engineering and Computer Sciences and the Department of Statistics at UC Berkeley.
Apr 19 · 16 min read

Photo credit: Peg Skorpinski

**Artificial Intelligence—The Revolution Hasn't Happened Yet**

Artificial Intelligence (AI) is the mantra of the current era. The phrase is intoned by technologists, academicians, journalists and venture capitalists alike. As with many phrases that cross over from technical academic fields

https://medium.com/@mijordan3/artificial-intelligence-the-revolution-hasnt-happened-yet-5e1d5812e1e7

How should we overcome these challenges?
Last year, Prof. Michael Jordan of UC Berkeley wrote a blog about the necessity of engineering discipline for machine learning. It is a long text, but my understanding is this:

As we have an engineering discipline called "civil engineering" for building bridges and buildings safely, we should have a similar engineering discipline for using machine learning or data analytics for decision making.

## The role of engineering

Why do we trust bridges?


Because of the accumulated knowledge called Civil Engineering

Theories（e.g., structure） ＊ **Safety Factor**

表-2.4 標準的な安全係数の値 (Standard values for safety factors)

| 安全係数 / 限界状態 | 材料係数 $\gamma_m$ コンクリート $\gamma_c$ | 材料係数 $\gamma_m$ 鋼材 $\gamma_s$ | 部材係数 $\gamma_b$ | 荷重係数 $\gamma_f$ | 構造解析係数 $\gamma_a$ | 構造物係数 $\gamma_i$ |
|---|---|---|---|---|---|---|
| 終局限界状態 | 1.3 | 1.0 | 1.15 ～ 1.3* | 1.0 ～ 1.2 | 1.0 ～ 1.2 | 1.0 ～ 1.2 |

Civil Engineering Handbook, p999

**New technology is accepted by the society only after it becomes engineering descipline**

What is engineering? Let's consider why we feel safe when crossing a bridge. It is because there is a huge body of knowledge called "Civil Engineering" behind the bridge, and we, society as a whole, trust it.

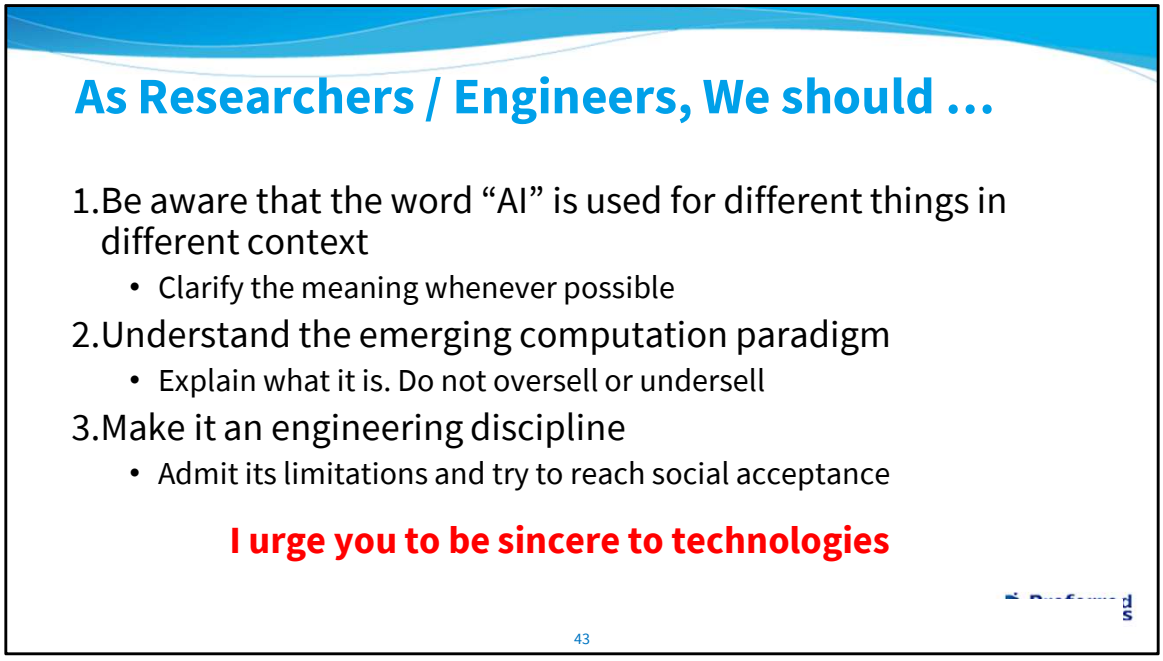
These volumes are Civil Engineering Handbook which have thousands of pages. What are in this body of knowledge? Most of it are theories such as material theory, structural theory, etc. In addition to these theories, there is a small set of numbers called "safety factors" shown in this table. For each subfield of civil engineering there is a safety factor, and if we put them altogether, the overall safety factor is around three. This means that, they first calculate a design of a bridge that is strong enough in every aspect of the relevant theories, and then finally they build actual bridge that is three times stronger than what the theories predict to be safe.

In other words, civil engineers are humble. They know that their theories are not perfect. There are things that are beyond their knowledge. The safety factors are determined according to their best judgement, and the society trusts them.

To me, engineering is a contract between the technology and the society.
We need something similar for the new computation paradigm.


MLSE (Machine Learning Systems Engineering) is a Special Interest Group of Japan Society for Software Science and Technology, established in April 2018.
We have a lot of discussions both domestically and globally on these topics.

We are planning to have MLSE 2019, a two-day, overnight workshop on 7/6-7/7 in Hakone. You are welcome to join. Paper submission deadline is 6/11.




Here is the take-away message of my talk today: as researchers and engineers in the field of AI, we should

1. Be aware that the word “AI” is used for different things in different context
    - Clarify the meaning whenever possible
2. Understand the emerging computation paradigm
    - Explain what it is. Do not oversell or undersell
3. Make it an engineering discipline
    - Admit its limitations and try to reach social acceptance

In short, I urge you to be sincere to technologies.


Thank You
Twitter: @maruyama
Preferred Networks
44