コンピュータサイエンスの役割

丸山宏
花王/東大/PFN
Twitter/X: @maruyama

自己紹介:

- 1975 高校のミニコンで初めてのプログラミング
- 1977 東京工業大学理学部情報科学科へ進学
  - 当時情報科学を専門に教える大学は少なかった
  - 修士課程:米澤明憲先生の指導で、自然言語理解システムの研究
- 1983-2009 日本IBM東京基礎研究所
  - 自然言語処理、機械翻訳、手書き文字認識、XML、Webサービス、セキュリティなどの研究・開発・標準化・コンサルティング
  - 最後の3年間は基礎研所長

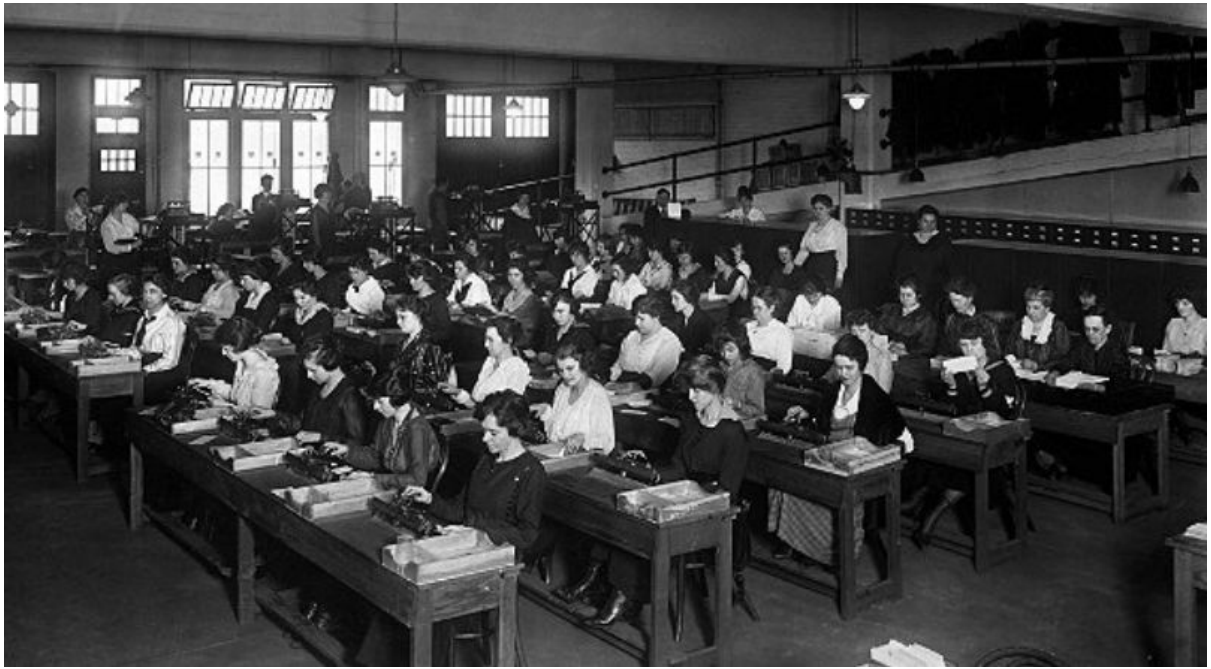
いわゆるコンピュータサイエンス

- 2011-2016 統計数理研究所
  - 統計モデリング、ビッグデータ解析、レジリエンス
- 2016年4月～ 株式会社Preferred Networks
  - 深層学習
- 現在:花王エグゼクティブフェロー／PFN取締役／東大客員教授

統計
データサイエンス
機械学習

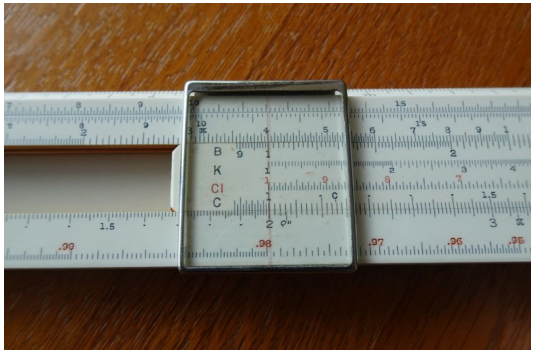
目次

「コンピュータ」とは？



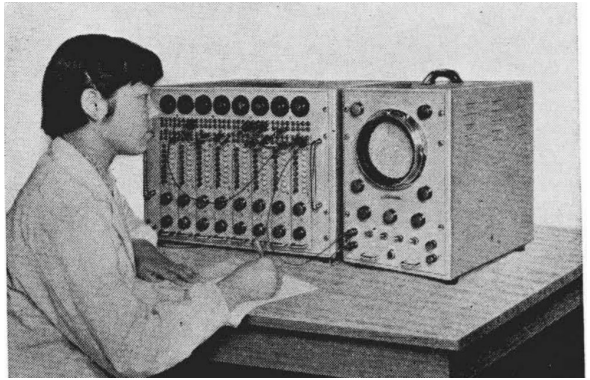
https://www.theatlantic.com/technology/archive/2013/10/computing-power-used-to-be-measured-in-kilo-girls/280633/

過去の計算機械の例



計算尺：スライドする目盛りの場所で表現

微分解析機：円盤回転角で表現
東京理科大近代科学資料館



日立ポータブルアナコン：
電圧・電流で表現
日立評論 Vol. 39, No. 2 (1957)

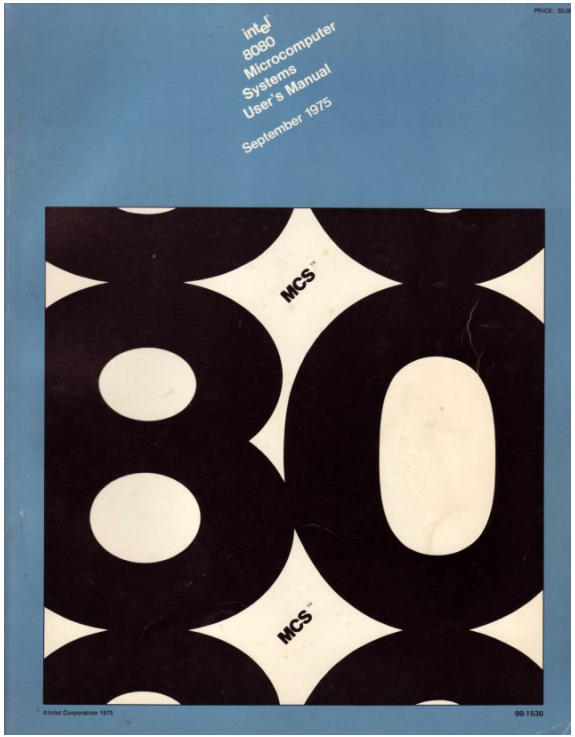
学問としての “Computer Science” の到来

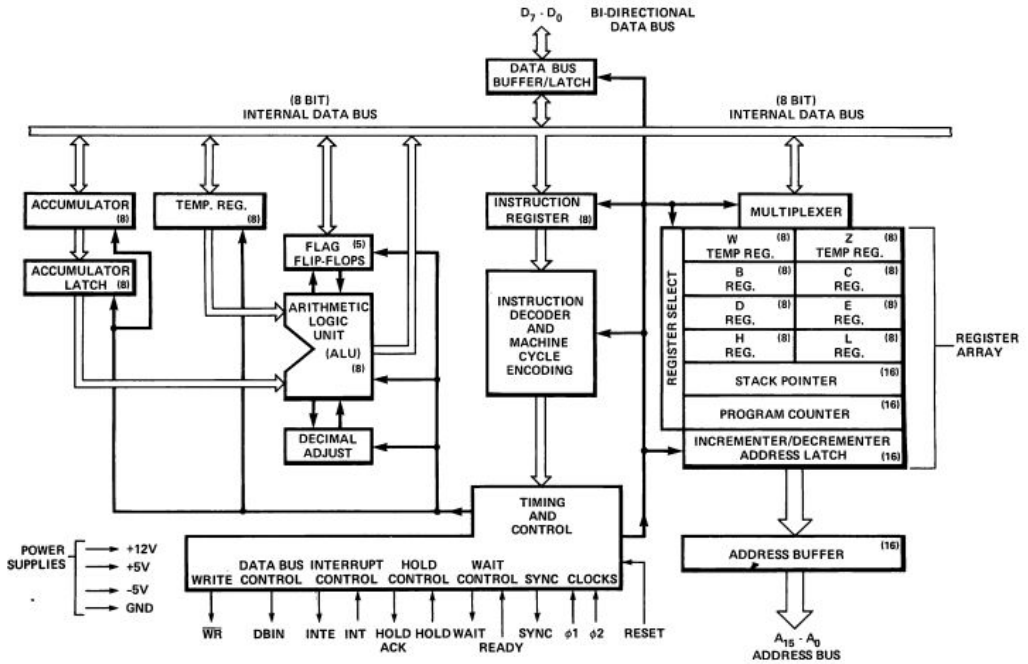
Thomas J. Watson Scientific Computing Laboratory, Feb. 1945

出典: Columbia大学,
http://www.columbia.edu/cu/computinghistory/612w116.html

フォン・ノイマン形式のデジタルコンピュータ

私が最初に学んだ計算機（紙上で、ですが…）

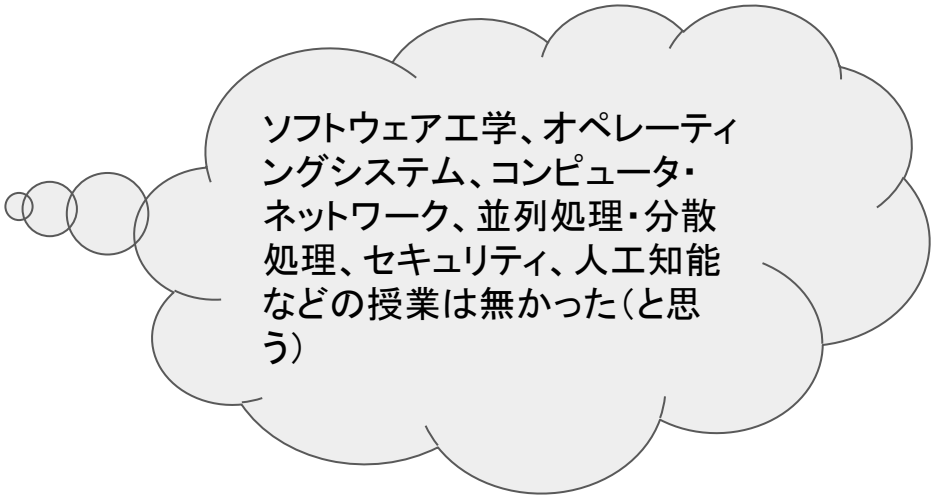




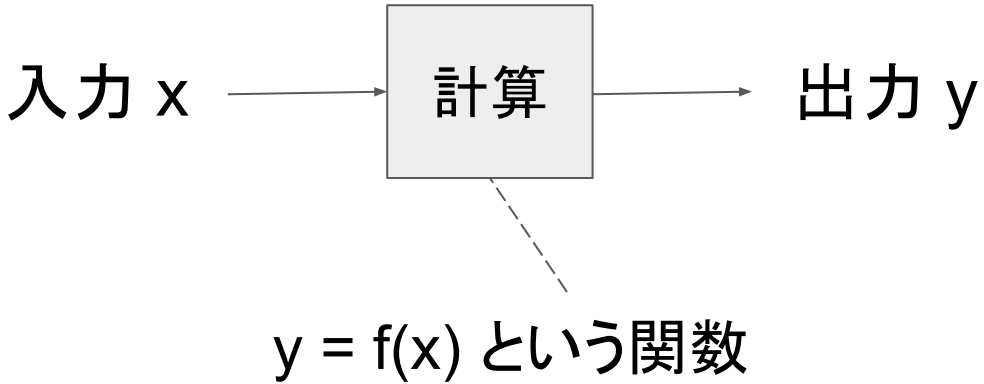
出典：intel 8080 Microcomputer Systems User’s Manual, Sept. 1975

1974-1977年に東工大情報科学科で習ったコンピュータ・サイエンス

- 計算の理論：原始帰納関数帰納関数・
  チューリングマシン・オートマトン・形式言語理論
- 集合・位相空間論
- 情報理論
- 計算量の理論
- 計算機アーキテクチャ
- プログラミング言語
- コンパイラ
- アルゴリズム・データ構造
- 数理計画法
- 確率・統計
- 数値解析法
- ：

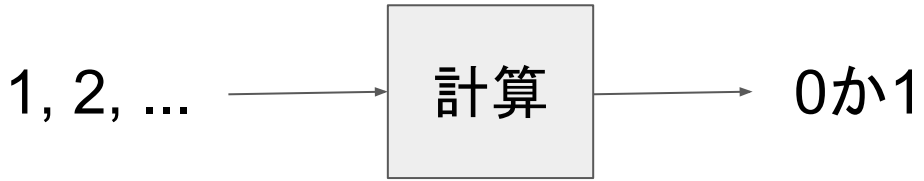


計算とは何か -- ここでは「関数」と考える



そもそも世の中はどのくらい複雑なのか？
自然数上での計算（デジタルな計算）はどのくらいあるか？

- 自然数　1, 2, 3, …
- 1つの自然数を0 か1に対応付ける関数（計算）を考えてみよう
  - （「自然数を2つに仕分けする関数」と考えてもよい）



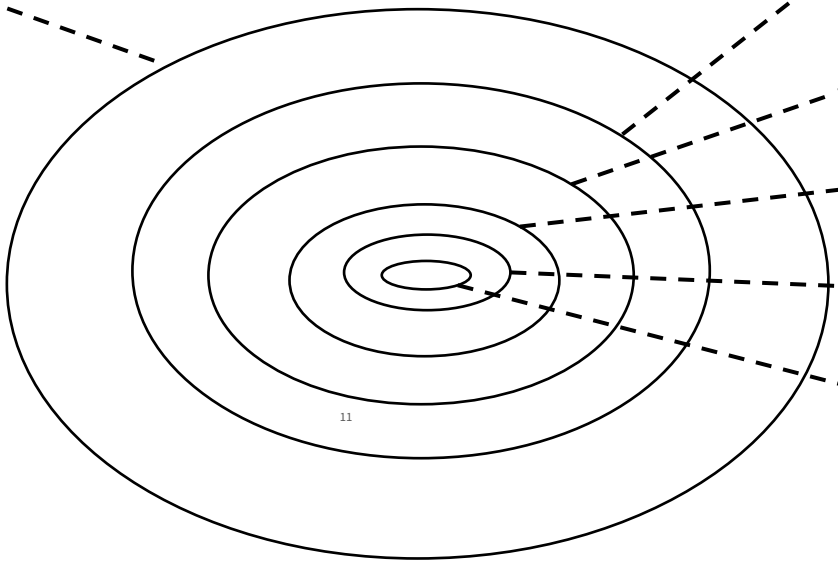
例：偶数だったら 0
　　奇数だったら 1

このような計算は非可算個ある
→「計算」には有限の記述ができないものもある！

CSにおける基礎理論の例：計算量階層

自然数から{0,1}への写像の全体

$$f : \mathbb{N} \to \{0,1\}, \quad f \in 2^{\mathbb{N}}, \quad |2^{\mathbb{N}}| = \aleph_1$$



計算可能関数（有限プログラムの全体）　$\aleph_0$

全域計算可能（必ず停止する）

P-SPACE

NP完全

P（普通のプログラム）

我々が普段扱う「計算」は極めて小さい領域

目次

1. デジタル計算機の理論としてのCS

2. 「計算」の理論としてのCS
   - 新しい計算の原理：深層学習と最適化
   - 仕様の与え方によるクラス分け

3. 社会基盤の理論としてのCS

4. 哲学的思考に資するCS

普通のプログラムの作り方: 例 摂氏から華氏への変換

要求

入力: C
出力: F
ただし、FはCを華氏で表したもの

モデル

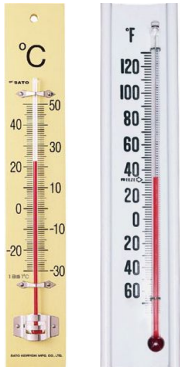
F = 1.8 * C + 32

アルゴリズム

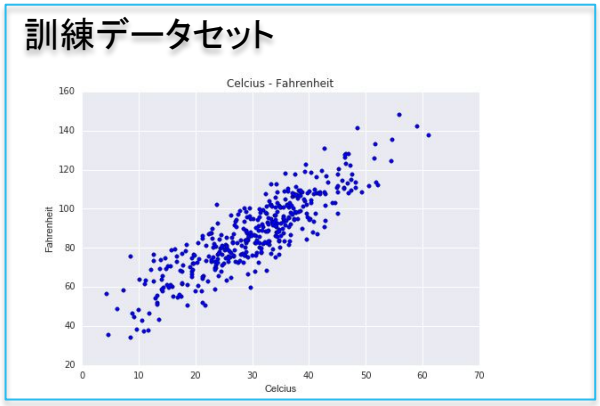
```
double c2f(double c) {
  return 1.8*c + 32.0;
}
```

モデルが既知／計算機構（アルゴリズム）が構成可能である必要

深層学習のやり方 – 訓練データで入出力を例示



観測



訓練（ほぼ自動でパラメタθを決定）

X → Y = f(X,θ) → Y

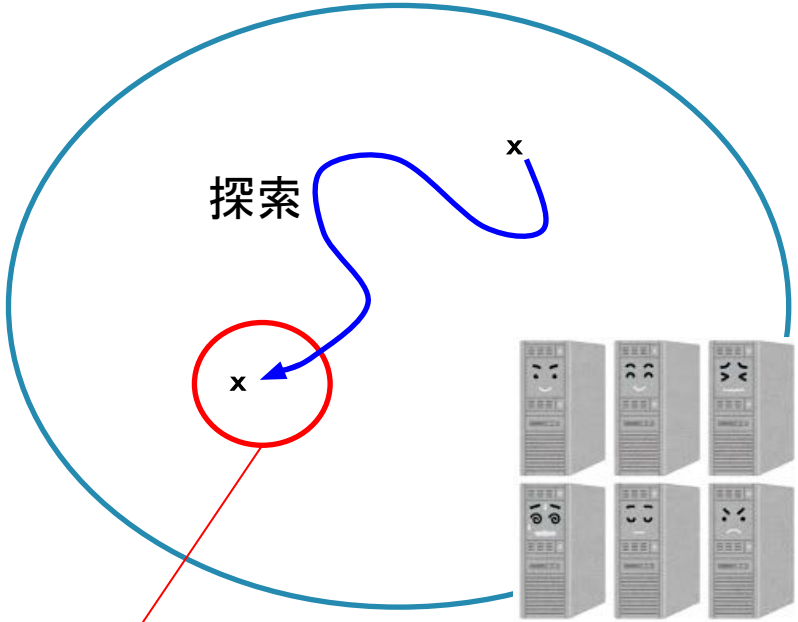
モデル／アルゴリズムが未知でもよい！

作るプログラム、探すプログラム

作るプログラム

```
def read_hourly_data(pattern, d, h):
    index_cols = patterns[pattern]
    d_str = d.strftime("%Y%m%d")
    d_h = d + pd.Timedelta('{} hours'.format
    fn = data_dir + "s3/realtime/{}/clipped_
      .format(d_str, d_str, h, pattern[0:5])
    print("Reading {}...".format(fn))
    if os.path.exists(fn):
        with open(fn, 'rb') as f:
            hourly = pd.read_csv(fn, usecols
            hourly = hourly.set_index(index_
            hourly.rename(columns={"populati
    else:
        hourly = pd.DataFrame(columns=index_
        hourly.set_index(index_cols,
        hourly.rename(colu
    return hourly
```

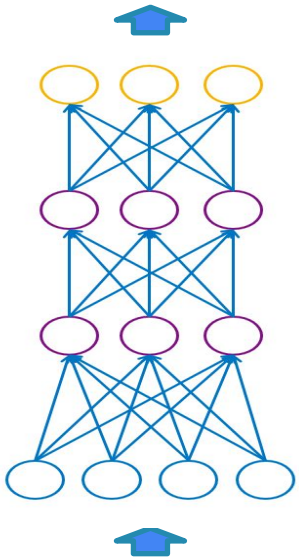
プログラム全体の空間



仕様Sを満たすプログラムたち

汎用計算機構としての深層学習

出力（超多次元）



入力（超多次元）

- 桁違いに多いパラメタ
  - 任意の多次元非線形関数を近似*
  - **疑似的にチューリング完全！**

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

深層学習によるプログラミング

- 仕様ではなく、訓練データセットを与える
- テストオラクルがない
- すべてが絡み合うCACE特性 -- “Change Anything Changes Everything”
- プログラム検証における、invariant が定義できない

- 形式的な仕様が書けない問題も（なんとなく）解ける
- 入力に関わらず、計算時間はほぼ一定
- 品質に関してはプロセスの評価ではなく、プロダクトメトリックが得られる

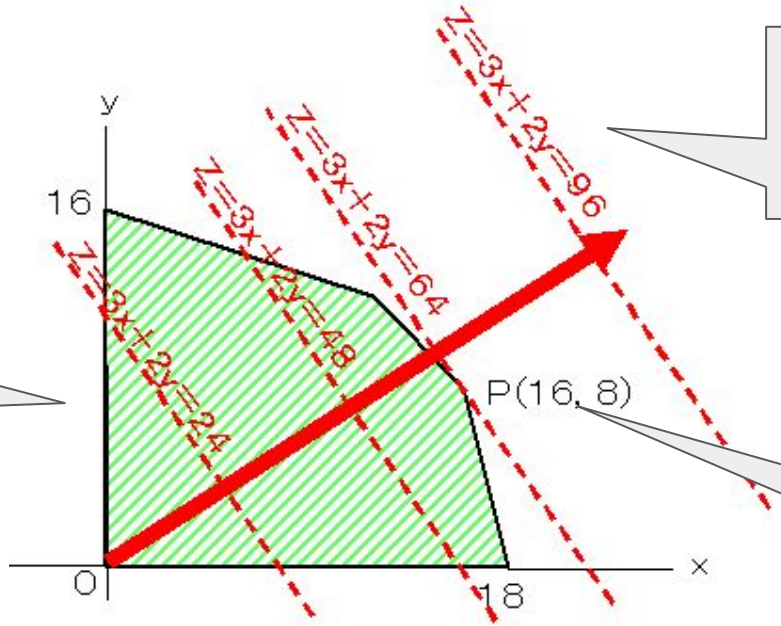
従来のプログラミングとは異なる ⇒ 機械学習工学

ISBN-13 ： 978-4065285862

忘れてはいけない最適化

人工知能のもう1つの流れ -- 解の探索（最適化）

例：“緑色の (x, y) 空間の中で、3x + 2y を最大にする x, y を求めよ”



効用関数が線形式で表される。

Uボートの攻撃成功確率に関する仮説

実行可能空間が線形不等式で表される。

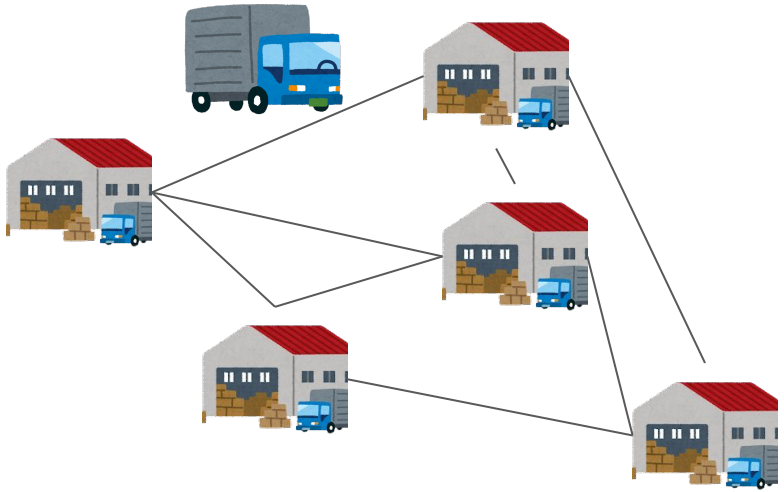
各商船・護衛艦の出発地・目的地・積荷・速度 …

見つかった**最適解**

出典: 小暮仁 Web教材「OR（オペレーションズ・リサーチ）」

離散領域における解の探索

- 定義域が離散の制約充足問題
  - 3充足問題など
- 定義域が離散の最適化問題
  - 巡回セールスマン問題
  - ナップザック問題



これらはいずれも**NP完全問題**として知られ、最悪の場合は2^n の時間がかかる

現在では経験的に効率のよいアルゴリズムがあり、ほとんどの現実問題は効率よく解ける

cf: https://twitter.com/maruyama/status/1620898201304969216

効用関数が事前にわからない問題設定 -- ブラックボックス最適化

効用関数 $\mu(x)$ の全体像はわからないが、ある点 x が与えられたとき $\mu(x)$ の値を返す**手続き**（オラクル）が先験知識として与えられている

- 多腕バンディット問題
- 実験計画法
- 進化計算
- 強化学習
- ベイズ最適化
- ：

期待値 $\mu(x_1)$　$\mu(x_2)$　$\mu(x_n)$

“期待値の不明なスロットマシンが n 台ある。
儲けを最大にするにはどうしたらよいか？ ”
（ $\mu(x_i)$ が事前にわかっていれば簡単！）

探索（exploration）と活用（exploitation）のバランスの取れた組み合わせが必要！

自身の探索行動により効用関数が変化する最適化 -- ゲーム理論

複数主体がそれぞれの利得最大化を目指して行動する状況における最適化

ユーザーの行動に基づいて期待値
を調整し売上を増大させる

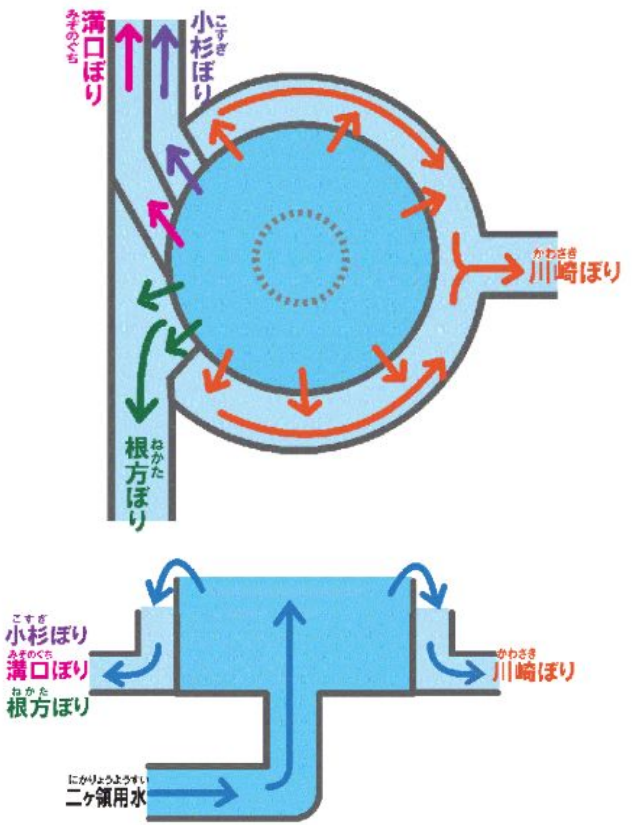
期待値 $\mu(x_1)$ $\mu(x_2)$ $\mu(x_n)$

相手の戦略を想定して戦略を考える

さらにエキゾチックな「計算」

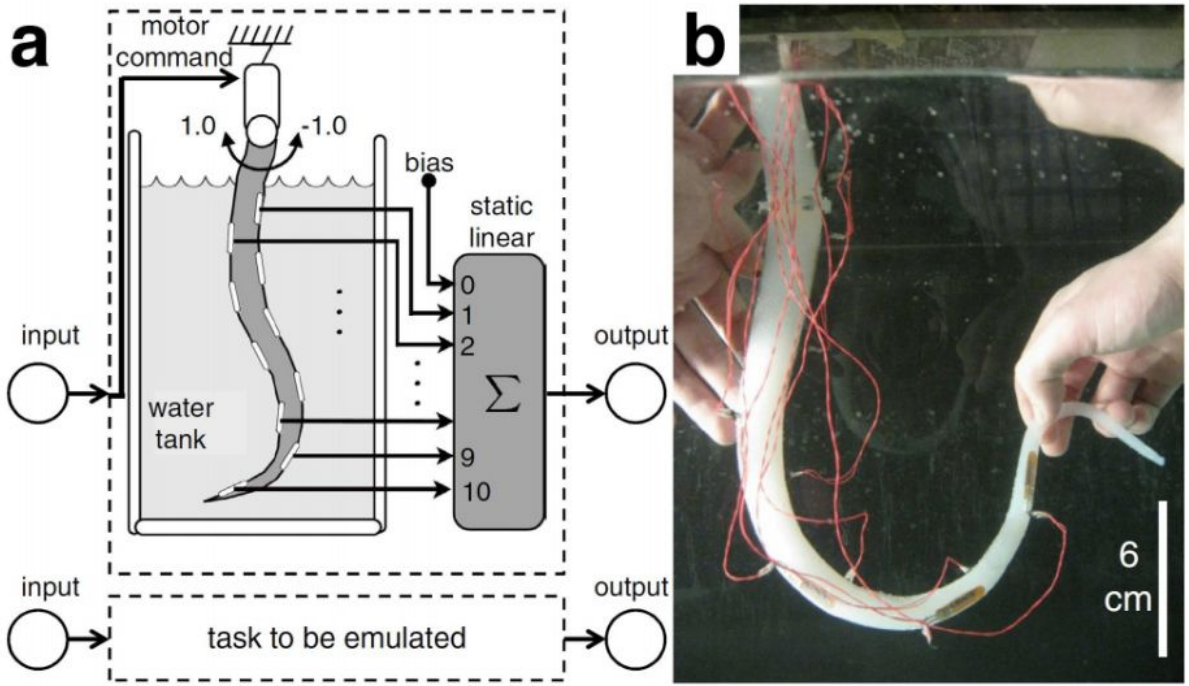
古典計算でない計算 (1) : 久地円筒分水

上流の水量に関わらず、水を常に7.415 (根方): 38.471(川崎) : 2.702 (溝口): 1.675(小杉)の割合で分けるには？



出典 : 川崎市ホームページ
http://www.city.kawasaki.jp/530/page/0000018473.html

古典計算でない計算 (2)：東大「タコの足」計算



出典：Nakajima, K., Hauser, H., Li, T. et al. Information processing via physical soft body. Sci Rep 5, 10487 (2015). https://doi.org/10.1038/srep10487

https://twitter.com/sina_lana/status/1135419503519469569

古典計算でない計算 (3): 波動計算

Wave-Based Neuromorphic Computing Framework for Brain-Like Energy Efficiency and Integration

Yasunao Katayama, *Senior Member, IEEE*, Toshiyuki Yamane, *Member, IEEE*, Daiju Nakano, *Member, IEEE*, Ryosho Nakane, *Member, IEEE*, and Gouhei Tanaka, *Member, IEEE*

***Abstract*—We present a framework of wave-based neuromorphic computing aiming at brain-like capabilities and efficiencies with nanoscale device integration. We take advantage of the unique nature of elastic nondissipative wave dynamics in both computations and IO communications in between as a means to natively implement and execute neuromorphic computing functions such as weighted sum in a spatiotemporal manner. Lower bound analysis based on a memory model and wave group velocity scaling is provided for conceptual evaluations.**

***Index Terms*—Neuromorphic computing, nanocircuits.**

I. Introduction

NEUROMORPHIC computing has been gaining attentions as CMOS technology scaling progresses and saturates. The advancement of CMOS technology allows for the integration of not only a limited number of neurons but a scale approaching that for realizing specific brain functions [1]–[4]. However, even back-of-the-envelope calculations show that significant gaps remain in constructing faithful human-brain scale

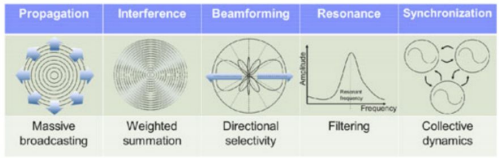
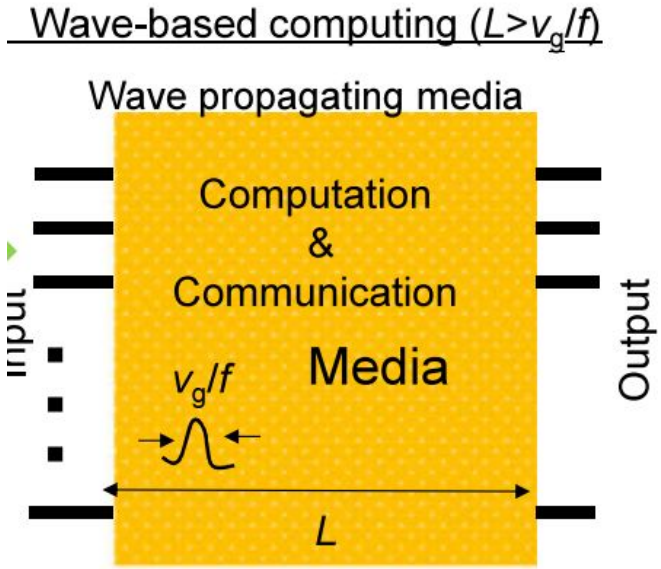


Fig. 1. Various aspects of wave characteristics for neuromorphic functions.

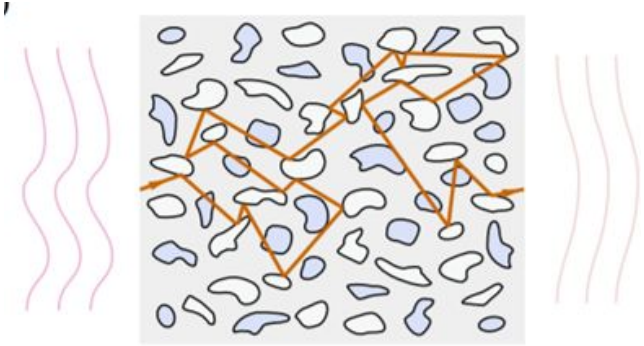
physical wiring limitations [6] and logical overheads for complicated collision and congestion controls in wire multiplexing techniques.

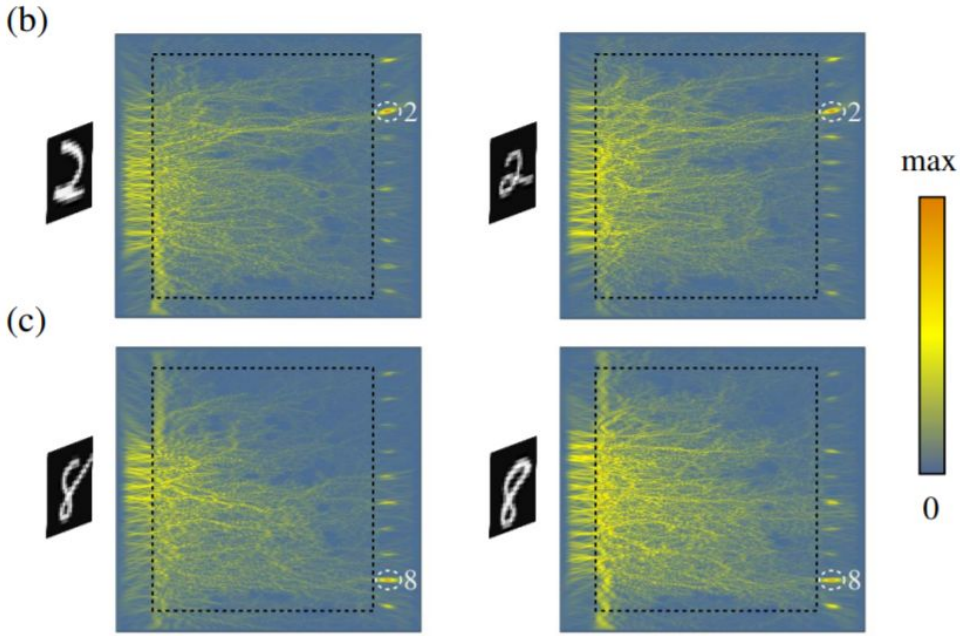
Meanwhile, wave-based computation frameworks have been studied as an alternative to conventional CMOS computing focusing on digital operations [7]–[9]. However, it is often a challenge to make the alternative superior to the present mature CMOS computing in conventional digital computing frameworks since standard digital CMOS is a very well designed bi-



出典: Y. Katayama, Wave-Based Neuromorphic Computing Framework for Brain-Like Energy Efficiency and Integration, IEEE Trans. on Nanotechonology, Vol. 15, No. 5, 2016.

古典計算でない計算 (4)：Wisconsin大学 ナノフォトニック計算





出典：E. Khoram, et. al, “Nanophotonic Media for Artificial Neural Inference”, arXiv:1810.07815v4

古典計算でない計算 (その他)

- 量子計算
- アナログ計算
- 分子計算
- :

いずれもチューリング機械抽象ではなさそう…

「計算」の再分類：私案

丸山の仮説:「Turingは偉すぎた」

2019年夏、「計算の未来と社会」ワークショップを開催

IBM天城ホームステッド

参加者の皆さん

計算の未来と社会

♡ 3

Hiroshi Maruyama
2023年9月5日 14:24

（2019年8月に、CNETに投稿した記事を加筆修正の上再掲するものです）

私たちが「計算」を考えるとき、現代のデジタル計算機の上でプログラムを実行することで行う計算を無意識に想像します。一方、深層学習やブラックボックス最適化の技術が進歩してきたことによって、明示的にアルゴリズムを記述しなくてもよい「計算」が実用的になってきました。「計算」とは何でしょうか。「計算」とは何かを広く捉えるとき、私達の社会が何にどのようなインパクトがあるでしょうか。私たちは何に備えていけばよいでしょうか。

https://note.com/hiroshi_maruyama/n/n9d57b601acfd

計算をどのように指定するか(1): 古典計算

- 計算の手順を指定する (プログラムを書く-- 普段やっているプログラミング)
- 離散空間上、チューリング機械抽象

ソートする計算の手順

```
def bubble_sort(arr):
    n = len(arr)
    if n <= 1:
        return arr
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
    return arr
```

出典：ChatGPT

計算をどのように指定するか(2): モデル化可能計算

- 入出力の厳密に数学的な関係を与えるが、計算手順は問わない
- 結果は近似かもしれない

ライブラリを使った計算(ソートする)

```
# リストをソートし、新しいソート済みリストを作成
sorted_list = sorted(my_list)
```

SQL文(ある従業員の年間の支払額)

```
SELECT SUM(s.salary_amount) AS total_salary
FROM salaries s
JOIN employees e ON s.employee_id = e.employee_id
WHERE e.employee_id = 'O12345'
  AND YEAR(s.salary_date) = YEAR(CURRENT_DATE)
```

出典：ChatGPT

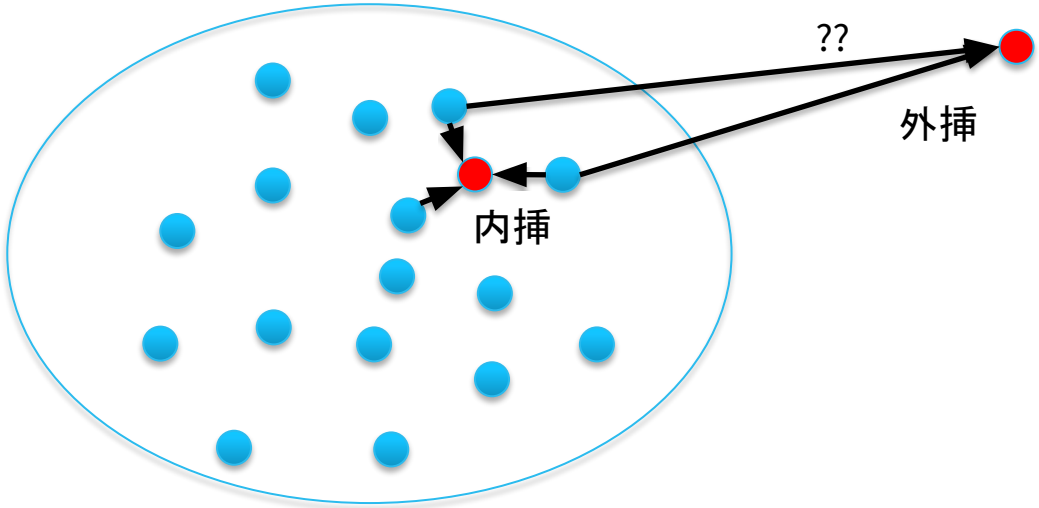
最適化((x-2)^2を最小化するxを求める)

```
import optuna

def objective(trial):
    x = trial.suggest_uniform('x', -10, 10)
    return (x - 2) ** 2

study = optuna.create_study()
study.optimize(objective, n_trials=100)

print(study.best_params)
```

出典：Optuna

計算をどのように指定するか(3): 部分再現可能計算

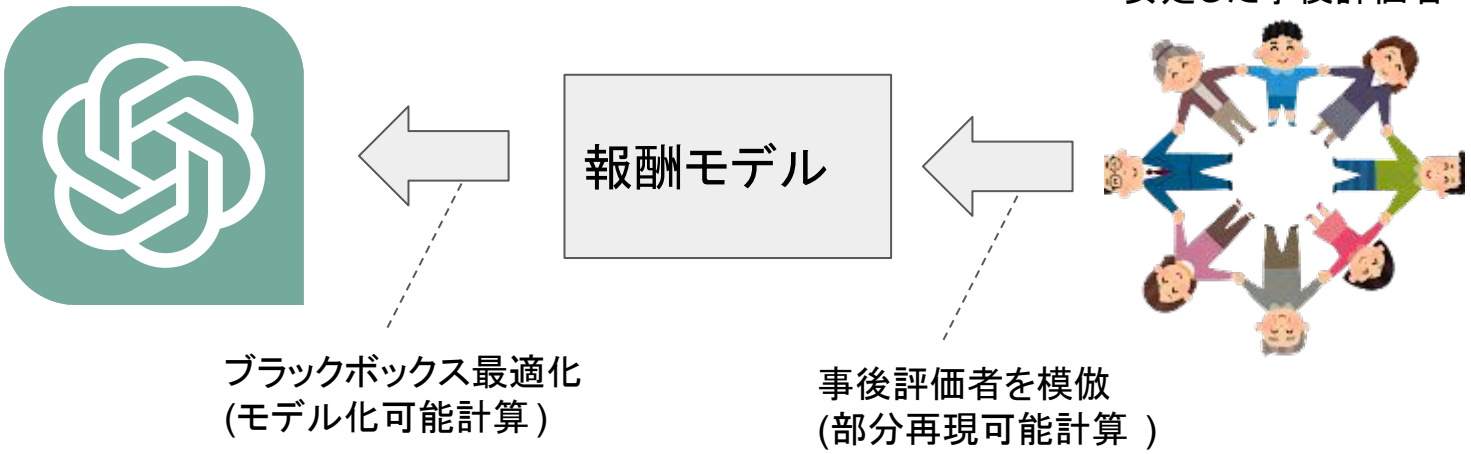
- 特定の入出力事例(訓練データセット)で指定する
- 入力が事例の近傍にあれば、その近似(内挿)を出力する
  - 例：帰納バイアス（汎化）を使って「良きに計らえ」
  - 非可算の計算を近似している可能性も …
- 入力が事例の近傍になければ、出力はあてにならない



計算をどのように指定するか(4): 事後評価可能計算

- 計算をやってみせないと、それが正しいかどうか言えない
- 安定した事後評価者をループに取り込めれば、ブラックボックス最適化問題(モデル化可能計算)に帰着できる

RLHF(人間のフィードバックによる強化学習)の例



仕様の与え方による「計算」のクラス分類

- 各クラスは、他クラスの計算では直接表現できないが、近似（エミュレート）することは(たぶん)できる

- それぞれのクラスによってツール・テクニックが異なる
  - 古典計算 -- アルゴリズム・データ構造・…
  - モデル化可能計算 -- 数学・論理学・ソルバー・…
  - 部分再現可能計算 -- 統計・機械学習工学・…
  - 事後評価可能計算 -- 社会学・アジャイル・…

- コンピュータサイエンスの課題
  - 各クラスの表現力は？ -- チューリング機械抽象でない、新しい計算の理論
  - 実装・ツール・テクニック -- 例: 機械学習工学
  - 1つのシステムは、上記4種の「計算」の組み合わせ
    - 「問題のどの部分をどの計算として表現するか」-- 新しいソフトウェア工学
  - :

目次

1. デジタル計算機の理論としてのCS

2. 「計算」の理論としてのCS

3. 社会基盤の理論としてのCS
    - 不確実性に対応するノウハウ

4. 哲学的思考に資するCS

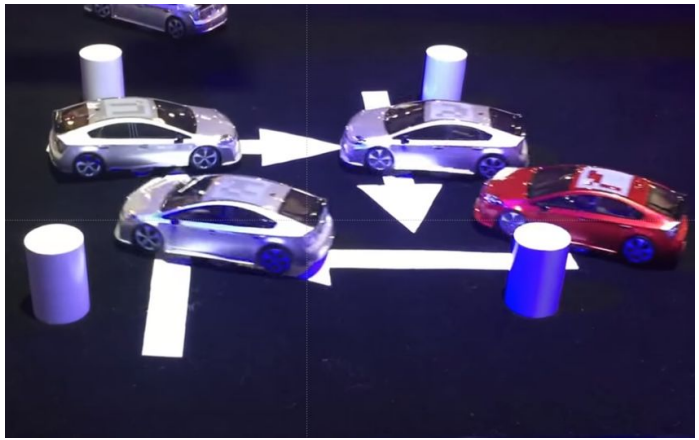
社会の基盤としてのIT

- 貨幣の仕組み → 電子マネー
- 流通 → Eコマース
- 教育 → Eラーニング
- 科学 → 高次元科学
- :

- 不確実性に対応するノウハウ

最適化において、効用関数を書き出すことの難しさ

強化学習において、衝突のペナルティを無限大にすると？



動かないクルマ

効用と安全性のバランスを定量的に要件として書き出す必要！

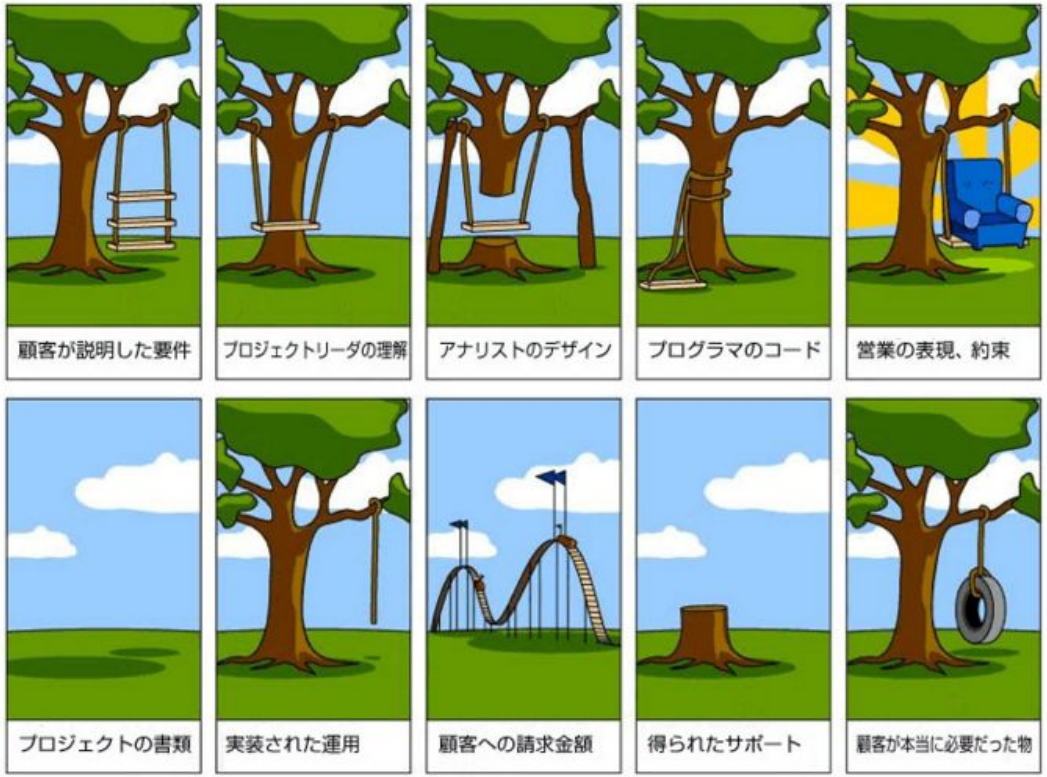
そもそも要求は「事前に」定義できるか -- 人工知能の未解決問題

- IJCAI 2017 Keynote by Stuart Russell, “Provably Beneficial AI”
  - 人:「コーヒーをとってきて」
  - ロボット: スタバへ行き、列に並んでいる他の客を殺してコーヒーをとってくる
  - 人の指示は常に不完全

多くの要求事項は、「後だしジャンケン」

人工知能研究での未解決問題:「フレーム問題」

要求定義の難しさはソフトウェア工学でよく知られた課題



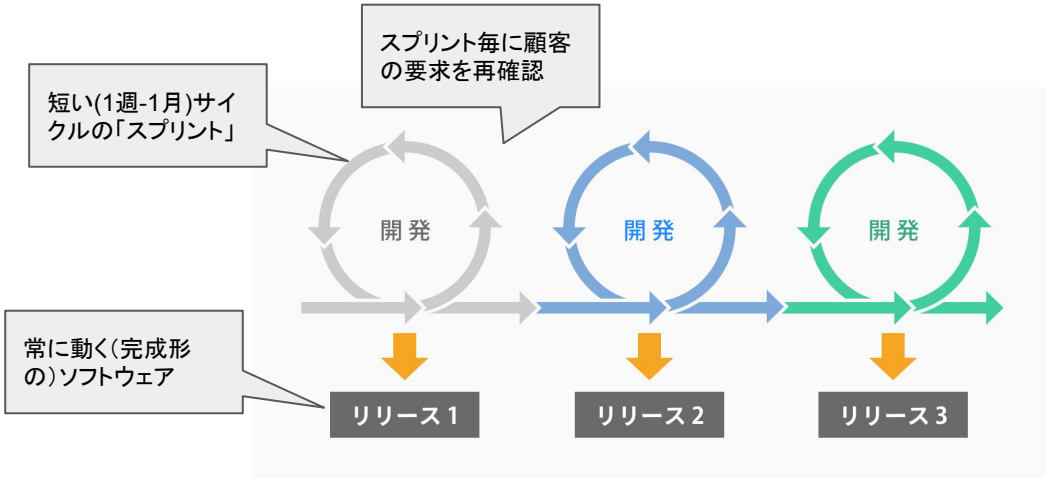
出展：Alexander C. 教授著、宮本雅明訳「オレゴン大学の実験」鹿島出版会(原題：The Oregon Experiment)

http://itpro.nikkeibp.co.jp/article/COLUMN/20080828/313626/

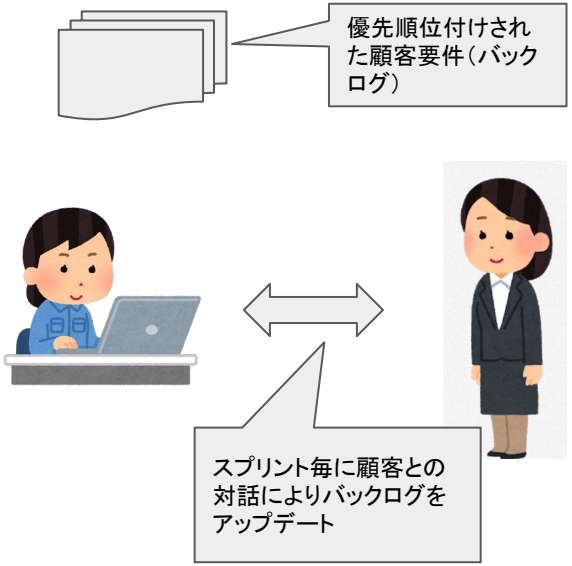
人は、自分が本当に欲しいものを表現するのが苦手！

ミダス王　出典：Wikipedia

要求の不確実性に対するソフトウェア工学の叡智：アジャイル開発



出典：https://backlog.com/ja/blog/what-is-agile-and-waterfall/



1. プロセスやツールよりも**個人と対話**を。
2. 包括的なドキュメントよりも**動くソフトウェア**を。
3. 契約交渉よりも**顧客との協調**を。
4. 計画に従うことよりも**変化への対応**を。

アジャイル・マニフェスト (2001)　https://agilemanifesto.org/

変化を事前に予測するのではなく、変化の事後に機敏に対応する

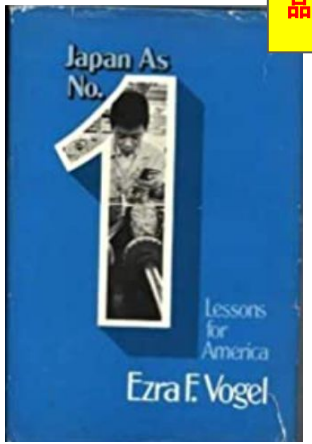
*QCD: Quality(品質), Cost, Delivery(納期)

組織論：なぜ多くの日本企業が苦しんでいるか（私見）

高度成長期

現場力
カイゼン
統計的品質管理

1979年



ISBN-13 ： 978-0674472150

1986 -
ISO 9000
品質マネジメント

2000
ISO 17799
セキュリティマネジメント

20年

ARTHUR ANDERSEN

Deloitte.
デロイト トーマツ

コーポレート・
ガバナンス

15年

ISBN-13 ： 978-4492396179

この間に何が起きたのか？

ガバナンスの対極にあるもの：アジャイル

ガバナンス

厳格なプロセス・ツール
包括的な文書
契約による縛り
詳細な計画と予実管理

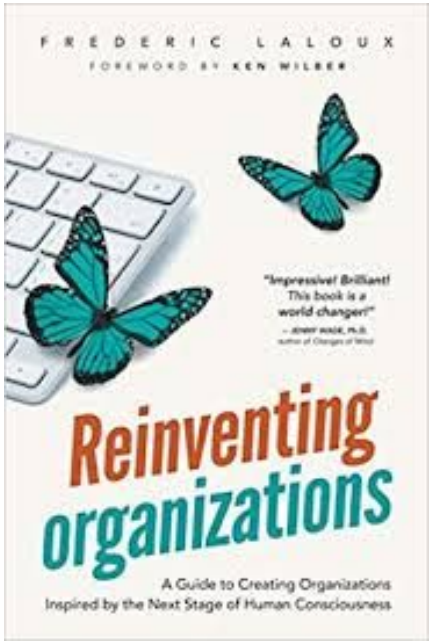
人は放っておくと何をするかわからない

アジャイル（エンパワメント）

個人との対話
動くソフトウェア（現場・現物）
顧客との協調（信頼）
変化への対応

信頼と協調

組織論の発展：自己組織型TEAL組織



出典：Frédéric Laloux, *Reinventing Organizations*, 2014.

オランダの訪問看護会社ビュートゾルフの例

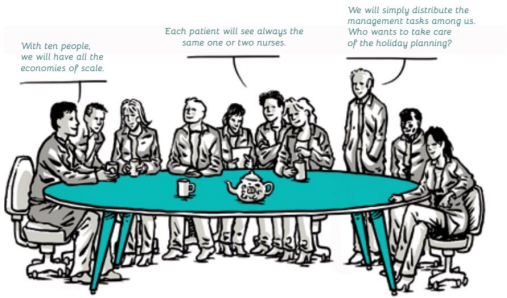
産業化され、効率化された訪問看護

10-12人のチームで自己管理

各自がその場で必要と思う看護を行う

出典：Frédéric Laloux, *Reinventing Organizations* , 2014.

ちょっと待てよ。これってQCサークルでは？



出典：Frédéric Laloux, *Reinventing Organizations* , 2014.

QCサークルとは、同じ職場内で品質管理活動を**自発的**に**小グループ**で行う活動

3つ基本理念

1. 人間の能力を発揮し、無限の可能性を引き出すこと。
2. **人間性を尊重**して、**生きがい**のある明るい職場をつくること。
3. 企業の体質改善・発展に寄与すること。

出典：Wikipedia

つまりは日本式カイゼンへの回帰ともいえる
アジャイルとルーツは同じ！

「ガバナンス」の神話

♡ 12

Hiroshi Maruyama
2023年8月2日 00:37

先日、東工大同窓会のイベントで、決めつけてはならないとき・決めなければならないときという講演をしました。いつの世も、意思決定は難しいものです。より良い意思決定をするにはどうしたらよいか、もちろんどこにも正解はないのですが、少なくともヒントとして使えそうなものはいくつかありそうです。その中で

- 自主管理型組織（TEAL型組織）
- アジャイル開発

https://note.com/hiroshi_maruyama/n/n2a04a9a7f8ee

目次

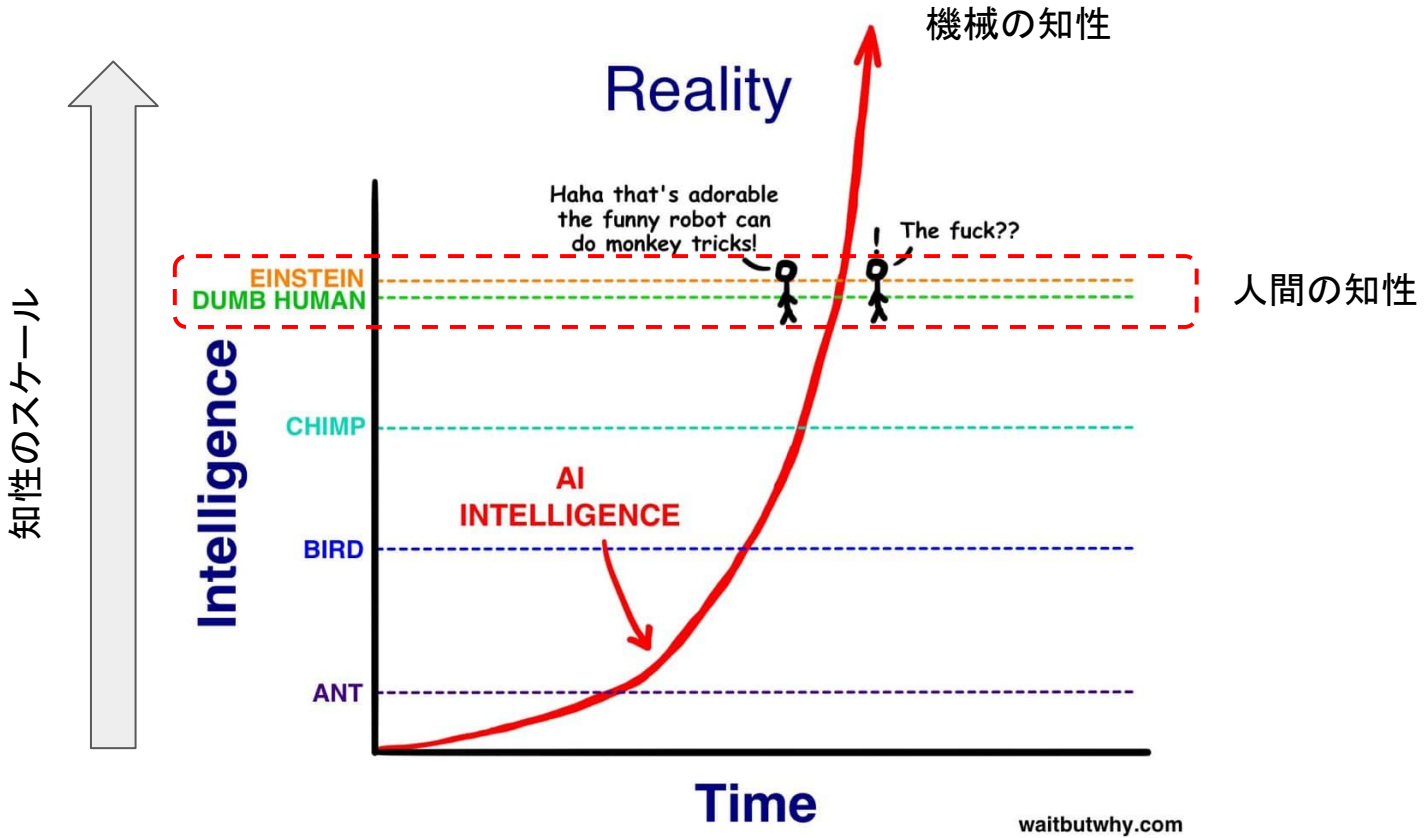
1. デジタル計算機の理論としてのCS

2. 「計算」の理論としてのCS

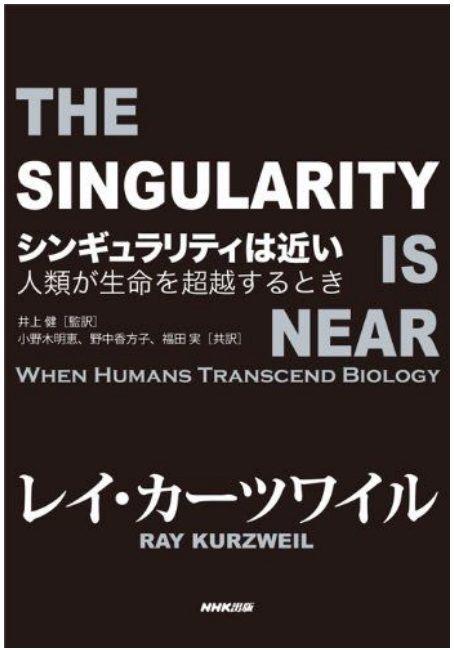
3. 社会基盤の理論としてのCS

4. 哲学的思考に資するCS
    - 知性とはなにか、人間とはなにか

人の知性 vs 機械の知性



レイ・カーツワイル「シンギュラリティは近い」



ASIN ： B009QW63BI, 2006

エポック1： 宇宙の生成（物理定数の決定）
エポック2： 生物の発生（DNAの情報）
エポック3： 脳の進化 （脳による情報処理）
エポック4： テクノロジ（情報処理ツール）
エポック5： 技術と脳の融合（**シンギュラリティ**）
エポック6： 宇宙の覚醒（宇宙全体の知性化）

収穫加速の法則

シンギュラリティ：
“When humans transcend biology with the power of technology”
(人類が技術によって生物の限界を超えるとき)

Singularity is here

（おそらく）

ChatGPTに関して、自民党AIプロジェクトチーム 安宅さんの資料より

ユヴァル・ノア・ハラリによる「人類の立ち位置」

“人類は自然界の中でなぜ特別な存在なのか ”

自然界の中で平等（アニミズム）

農耕の発明

「神の子」としての特別な存在

科学の発展

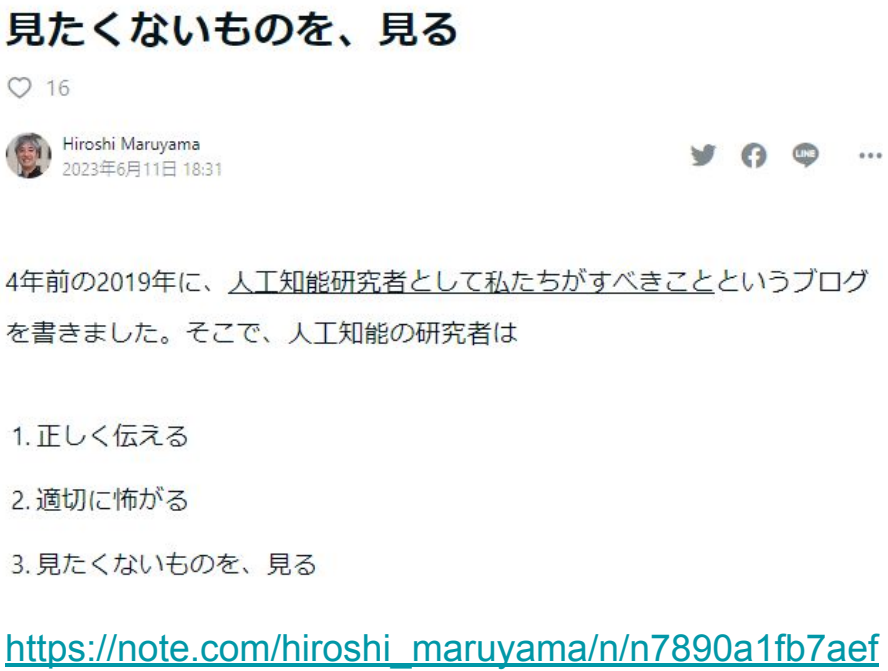
「考える存在」としての優位性（人間至上主義）

情報技術

データの中に埋没する存在（「データイズム」）

あらゆる面で人間より知的に優れた機械ができたとき、
人類は自然界の中で特別な存在であることを止めるのか？

相対化する知性



見たくないものを、見る

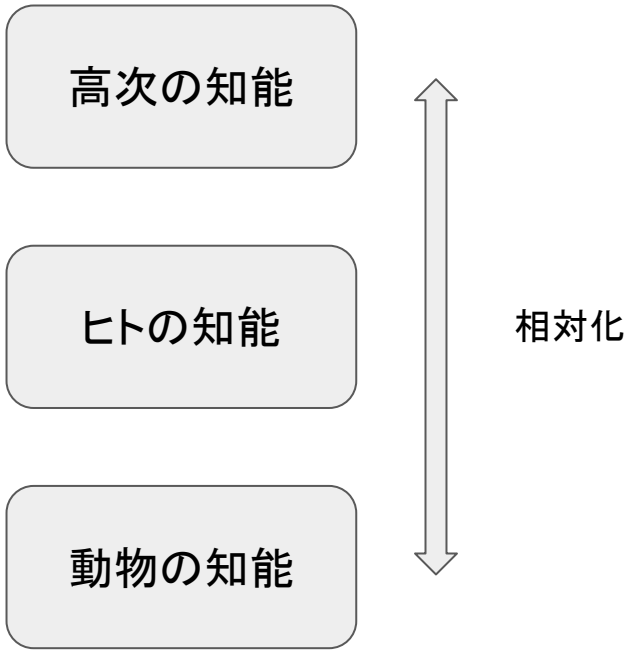
♡ 16

Hiroshi Maruyama
2023年6月11日 18:31

4年前の2019年に、人工知能研究者として私たちがすべきことというブログを書きました。そこで、人工知能の研究者は

1. 正しく伝える
2. 適切に怖がる
3. 見たくないものを、見る

https://note.com/hiroshi_maruyama/n/n7890a1fb7aef



私たちの将来：
Contain（閉じ込め）か　　Embrace（抱擁）か

Contain：機械に制約を課し、人間に対する脅威とならないようにする

- EU「包括的AI規制法案」
- アシモフ「ロボット三原則」

- 漏れのないポリシーは定義不可能かもしれない
- ポリシーを守らない人が出てくるかもしれない
- 地球外文明が、人間中心主義を脅威とみなすかもしれない

Embrace：機械との一体化により、人類文明全体として高次の存在を目指す

- カーツワイル「シンギュラリティ」
- A.C.クラーク「幼年期の終わり」

- 個人のアイデンティティは曖昧になるかもしれない
- いろいろな意味で「人間らしさ」を失うかもしれない
- 「基本的人権」を見直す必要があるかもしれない

まとめ

- CSの対象の拡大
  - チューリング機械でない計算原理
- CSでの知見の他分野への展開
  - 社会学・哲学

コンピュータ・サイエンスを広く捉えよう！

Thank You

@maruyama on twitter

ISBN-13 : 978-4764906068