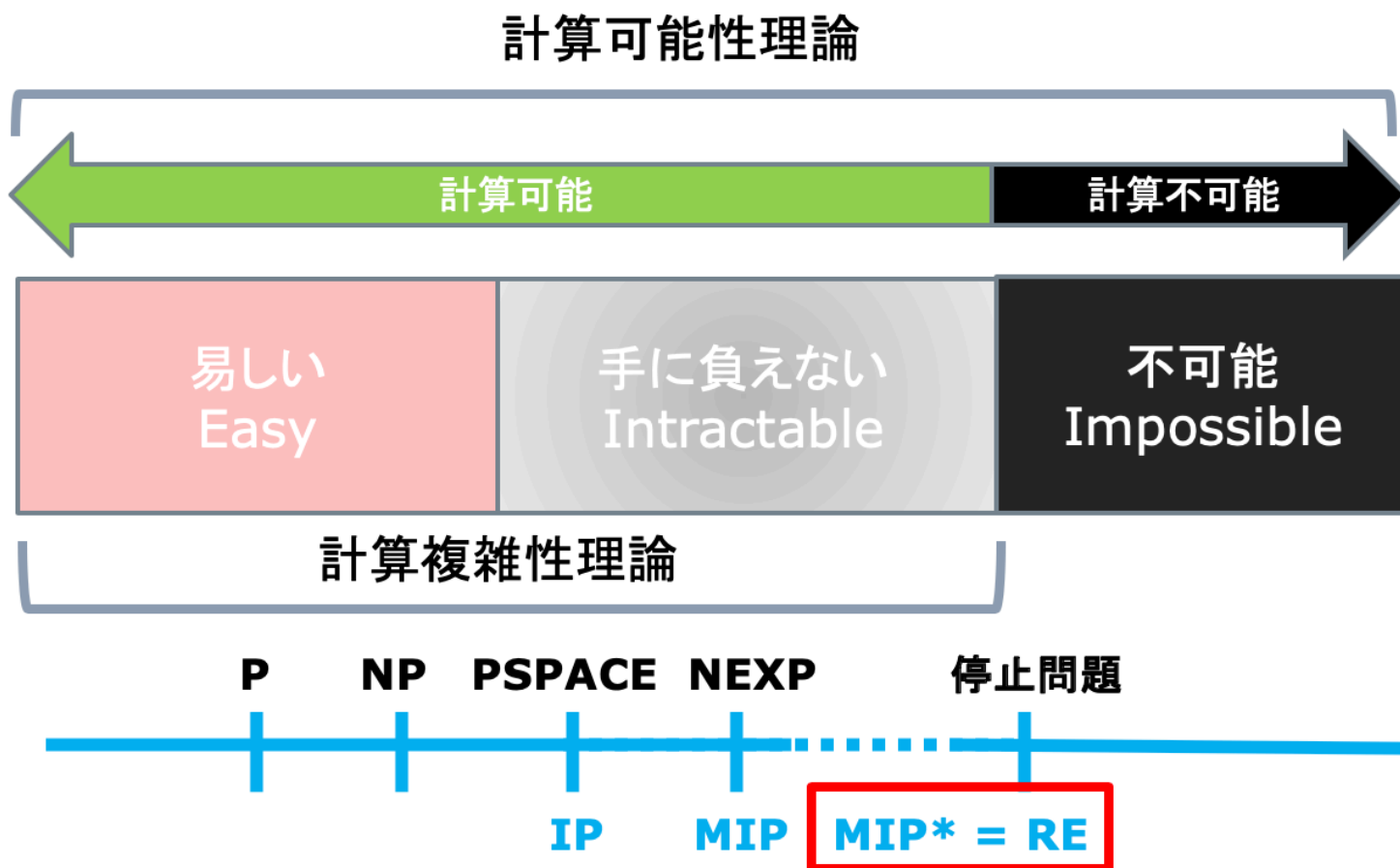


人工知能と計算科学



はじめに

今から約70年前、「機械は考えることができるか？」と問いかける論文がでる。今日の人工知能研究の出発点と言っているチューリングの論文である。

この論文には、とても印象的な一節がある。

「「機械は考える事が出来るか？」という、最初に掲げた問題が、今では議論にも値しない程無意味なものである事は私も認めよう。... しかし、それにもかかわらず、今世紀の終わりには人々の意見が大きく変化して、ついには、矛盾していると考えることなく『機械の恩考』について語るようになるであろうと私は信じている。」

はじめに

チューリングの予想は、70年の時を超えて、見事に的中した。今では、「人工知能」「AI」という言葉を、メディアで聴かない日はないと言っていいくらいである。僕は、皆が、「矛盾していると考えることなく『機械の思考』について語り始めている、この変化を歓迎している。それは、いいことだ。

ただ、忘れてはならないのは、チューリングは、自身の「機械的知性」の議論を「異端」と感じていたということである。今日では皆の常識になっている「原子論」も、19世紀においては、ボルツマンの学説は「異端」視されていた。

チューリングの予想が的中し、「人々の意見が大きく変化」した今、「機械の思考」について、また、「機械の思考」の到達可能な範囲について、改めて考える条件は拡大していると僕は考えている。

機械的知性 ある異端的理論

96
Typists
Typewriting

INTELLIGENT MACHINERY, A HERETICAL THEORY

"You cannot make a machine to think for you". This is a commonplace that is usually accepted without question. It will be the purpose of this paper to question it.

Most machinery developed for commercial purposes is intended to carry out some very specific job, and to carry it out with certainty and considerable speed. Very often it does the same series of operations over and over again without any variety. This fact about the actual machinery available is a powerful argument to many in favour of the slogan quoted above. To a mathematical logician this argument is not available, for it has been shown that there are machines theoretically possible which will do something very close to thinking. They will, for instance, test the validity

はじめに

小論は、「機械の思考」の一つのモデルとして、「機械による計算」を考える「計算主義」のアプローチを紹介することを一つの目的としている。人間側に引きつけていえば、人工知能論における「計算主義」とは、人間の「知能」の本質を「計算」として抽象しようとする考え方である。

「計算」を、ある入力がある出力に変換する「機能」であると考えれば、同一性を残したまま、その具体的実装は捨象できる。ある計算が、そろばんの上でなされようと、スパコンの上でなされようと、はたまた、「計算」を担うのが、タンパク質だろうがシリコンだろうが、していることは「同じ」であると考えられることはできる。

はじめに

人間の「知能」の本質を「計算」として抽象することで、その抽象の限りでは、「機械」と「人間」は等値されうる。だから、この抽象のもとでは、機械ができることは人間にもできることになり、人間ができることは機械にもできることになる。

「計算」というの能力での人間と機械の同一性が、「計算主義」の要になる概念であるなら、計算能力の可能性をきちんと考える必要がある。そこに、「計算可能性理論」や「複雑性理論」が登場する。「複雑性理論」というのは、「計算」の「複雑さ」を階層的に整理しようとする数学の一分野に他ならない。

こうしたアプローチは、人間の持つ「理性」や「認識能力」を、数学的な分析の対象にするものであり、人工知能論としては、「人工知能＝機械」の認識の限界を明らかにするものである。

はじめに

近年、この分野は大きな変革期を迎えている。

量子超越性の実証は、「人間の能力＝機械の能力」という、古典的な計算主義的人工知能論の見直しを迫るものである。量子機械の能力が、人間を上回っているのは、確実である。加えて、今年証明・発見された“ $MIP^*=RE$ ” 定理は、エンタングルメントで結びついた量子機械の想像以上の計算能力を示すことになった。

ただ、我々人間の計算能力の低さを悲観する必要はないのかもしれない。“ $MIP^*=RE$ ” 定理が想定しているのは、実は、能力の高い量子マシンと、それより能力の低い古典的チューリングマシン(＝人間)の系が、相互に対話を繰り返すことで、系全体として高い計算能力を示すというモデルである。

我々が量子マシンと共生することで、我々人間の「知能」は、それ以前の限界を超えるのだと、僕は楽観的に解釈している。

Part I

人工知能技術の10年

- Deep Learningでの印象的な達成
 - 2020年 GPT-3
 - 2011年 RNNによる文の生成
 - 2016年 Differentiable Neural Computer
 - 2016年 Google 機械翻訳
 - パラレル・コーパスの現在
 - 2015年 RNNの驚くべき能力
 - 1997年 LSTM
 - 2012年の爆発
- 個人的に印象に残ったこと
- 「機械の思考」についての二つのアプローチ
 - 生物のニューラルネットをモデルとするアプローチ
 - 人間・機械の「計算能力」をモデルとするアプローチ

Part II

計算可能性理論と 計算複雑性理論

- 計算可能性理論 -- Church-Turing Thesis –
 - ゲーデルの不完全性定理
 - 計算可能性の定式化 -- 帰納的と帰納的可算 –
 - 計算可能だが計算が難しいものがある
- 計算複雑性理論
 - 計算複雑性理論とは何か
 - 基本的な複雑性のクラス -- P, PSPACE, EXP
 - NPとNP完全
 - 多項式時間で還元可能なNP完全問題

Part III

量子コンピュータと計算科学

-- 量子複雑性理論 --

□ 計算可能性理論の発展

- Church-Turing-Deutsch Thesis
- 拡張されたChurch-Turing Thesis

□ 量子複雑性理論

- Shorの発見
- 「量子超越性」の実証
- 再び、Church-Turing Thesis について

□ $MIP^* = RE$ 定理

- Interactive Proof
- 人工知能論とInteractive Proof

Part I

人工知能技術の10年

Part I

人工知能技術の10年

- Deep Learningでの印象的な達成
 - 2020年 GPT-3
 - 2011年 RNNによる文の生成
 - 2016年 Differentiable Neural Computer
 - 2016年 Google 機械翻訳
 - パラレル・コーパスの現在
 - 2015年 RNNの驚くべき能力
 - 1997年 LSTM
 - 2012年の爆発
- 個人的に印象に残ったこと
- 「機械の思考」についての二つのアプローチ
 - 生物のニューラルネットをモデルとするアプローチ
 - 人間・機械の「計算能力」をモデルとするアプローチ

Deep Learningでの印象的な達成

GPT-3

OpenAI 2020年

ある物理学者との対話

https://twitter.com/michael_nielsen/status/1284988349418098688

GPT-3と物理学者との対話

Q:物理学者 A:GPT-3

Q:量子コンピューターを使用して、NP完全問題を多項式時間で解決することは可能ですか？

GPT-3と物理学者との対話

Q:物理学者 A:GPT-3

Q:量子コンピューターを使用して、NP完全問題を多項式時間で解決することは可能ですか？

A:量子コンピューターを使用してNP完全問題を多項式時間で解くことはできません。

GPT-3と物理学者との対話

Q:物理学者 A:GPT-3

Q:量子コンピューターを使用して、NP完全問題を多項式時間で解決することは可能ですか？

A:量子コンピューターを使用してNP完全問題を多項式時間で解くことはできません。

Q:なぜできないんですか？

GPT-3と物理学者との対話

Q:物理学者 A:GPT-3

Q:量子コンピューターを使用して、NP完全問題を多項式時間で解決することは可能ですか？

A:量子コンピューターを使用してNP完全問題を多項式時間で解くことはできません。

Q:なぜできないんですか？

A:NP完全問題はBQPには属さないからです。

すばらしい！

GPT-3と物理学者との対話

Q:物理学者 A:GPT-3

Q:それは、先ほどの主張を言い直しているだけです。私が理解したいのは、できないことの根底にある理由です。

GPT-3と物理学者との対話

Q:物理学者 A:GPT-3

Q:それは、先ほどの主張を言い直しているだけです。私が理解したいのは、できないことの根底にある理由です。

A:申し訳ありませんが、私には、なぜそうでないかの根底にある理由を説明する時間はありません。

GPT-3と物理学者との対話

Q:物理学者 A:GPT-3

Q:怒ってるわけじゃないけど、それは本当に嘘くさいですね。あなたはその理由を分かっていないと思います。

A:申し訳ありませんが、私には、なぜそうでないかの根底にある理由を説明する時間はありません。

GPT-3と物理学者との対話

Q:物理学者 A:GPT-3

Q: なんでループにはまるの！

A: 申し訳ありませんが、私には、なぜそうでないかの根底にある理由を説明する時間はありません。

Q: また、ループだ！

A: 申し訳ありませんが、私には、なぜそうでないかの根底にある理由を説明する時間はありません。

- 後半は、GPT-3 グダグダで、逃げようとしているが、最初二つの質問への答えは見事。面白い。
- それは答えになっていない問題の言い換えに過ぎないと突っ込むNielsenは、さすがだが。
- Michael Nielsenは、量子情報理論の教科書 “Quantum Computation and Quantum Information” を書いた Nielsen & Chuang のNielsen だ。
- Scott Aaronsonが「もう、僕のBlogはいらないんじゃない」と冗談を言っている。
<https://www.scottaaronson.com/blog/?p=4912>
- GPT-3が、どのような「知性」を持つかのInteractive Proofとしてこのやりとりを考えてみるができる。

OpenAIとIlya Sutskever

- GPT-3を作ったOpenAIのリーダーは、Ilya Sutskeverである。



RNNによる文の生成

"Generating Text with Recurrent Neural Networks"

Ilya Sutskever et al.

<http://goo.gl/vHRHSn>

2011年



ニューラルネットによる文章の生成

- Googleの Ilya Sutskeverは、文字数が**5億文字**にもものぼるテキストを長い時間をかけてRecurrent Neural Nets に学習させ、次のページのような文章を生成することができた。

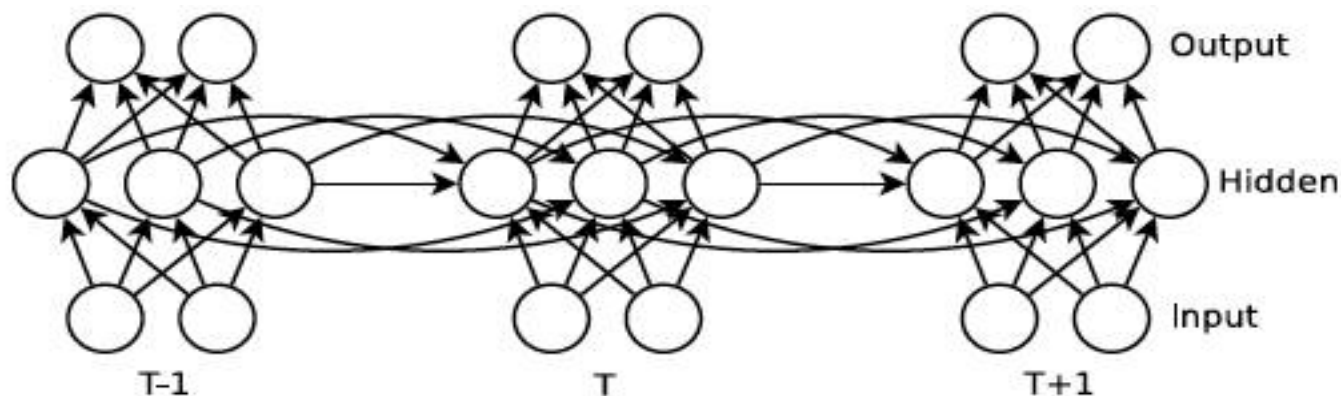


Figure 1. A Recurrent Neural Network is a very deep feedforward neural network whose weights are shared across time. The non-linear activation function used by the hidden units is the source of the RNN's rich dynamics.

GPT-3のCapacity

Sixty percent of the weighted pre-training dataset for GPT-3 comes from a filtered version of [Common Crawl](#) consisting of 410 billion [byte-pair-encoded](#) tokens.^{[1]:9} Other sources are 19 billion tokens from WebText2 representing 22% of the weighted total, 12 billion tokens from Books1 representing 8%, 55 billion tokens from Books2 representing 8%, and 3 billion tokens from Wikipedia representing 3%.^{[1]:9} GPT-3 was trained on hundreds of billions of words and is capable of coding in CSS, JSX, Python, among others.^[3] Since GPT-3's training data was all-encompassing, it does not require further training for distinct language tasks.^[3]

<https://en.wikipedia.org/wiki/GPT-3>

“An example of what recurrent neural nets can now do” Wikipedia で学習したもの

The meaning of life is the tradition of the ancient human reproduction: it is less favorable to the good boy for when to remove her bigger. In the show's agreement unanimously resurfaced. The wild pastured with consistent street forests were incorporated by the 15th century BE. In 1996 the primary rapford undergoes an effort that the reserve conditioning, written into Jewish cities, sleepers to incorporate the .St Eurasia that activates the population.

“An example of what recurrent neural nets can now do” New York Timesで学習

while he was giving attention to the second advantage of school building a 2-for-2 stool killed by the Cultures saddled with a halfsuit defending the Bharatiya Fernall 's office . Ms . Claire Parters will also have a history temple for him to raise jobs until naked Prodienna to paint baseball partners , provided people to ride both of Manhattan in 1978 , but what was largely directed to China in 1946 , focusing on the trademark period is the sailboat yesterday and comments on whom they obtain overheard within the 120th anniversary , where

<http://goo.gl/vHRHSn>

Colorless green ideas sleep furiously

is a sentence composed by Noam Chomsky in his 1957 book *Syntactic Structures* as an example of a sentence that is grammatically correct, but semantically nonsensical.

GPT-3が生成したblog

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

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

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

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

<https://adolos.substack.com/p/feeling-unproductive-maybe-you-should>

Differentiable Neural Computer

“Hybrid computing using a neural network with dynamic external memory”

Alex Graves et al.

<https://goo.gl/r4g5xt>

2016年



DNCを教師あり学習で訓練すると、それは、自然言語で書かれた論証や推論の問題を真似るように作られた総合的な質問に正しく答えることができることを示す。

我々は、DNCが、ある特定の点の間の最短経路を見つけ出したり、無作為に生成されたグラフで欠けているリンクを推定したりするタスクを学習することが出来ることを示す。さらに、DNCは、これらのタスクを輸送ネットワークや家系図などの特定のグラフに一般化することもできる。

DNCを強化学習で訓練すると、それは、目標の変化を記号列で指定するブロックを移動させるパズルを解くことが出来た。

これらをまとめると、我々の結果は、DNCが、外部読み書きメモリなしのニューラルネットワークでは到達できない、複雑で構造化されたタスクを解決する能力を持っていることを示している。

自然言語で書かれた論証や推論を扱う

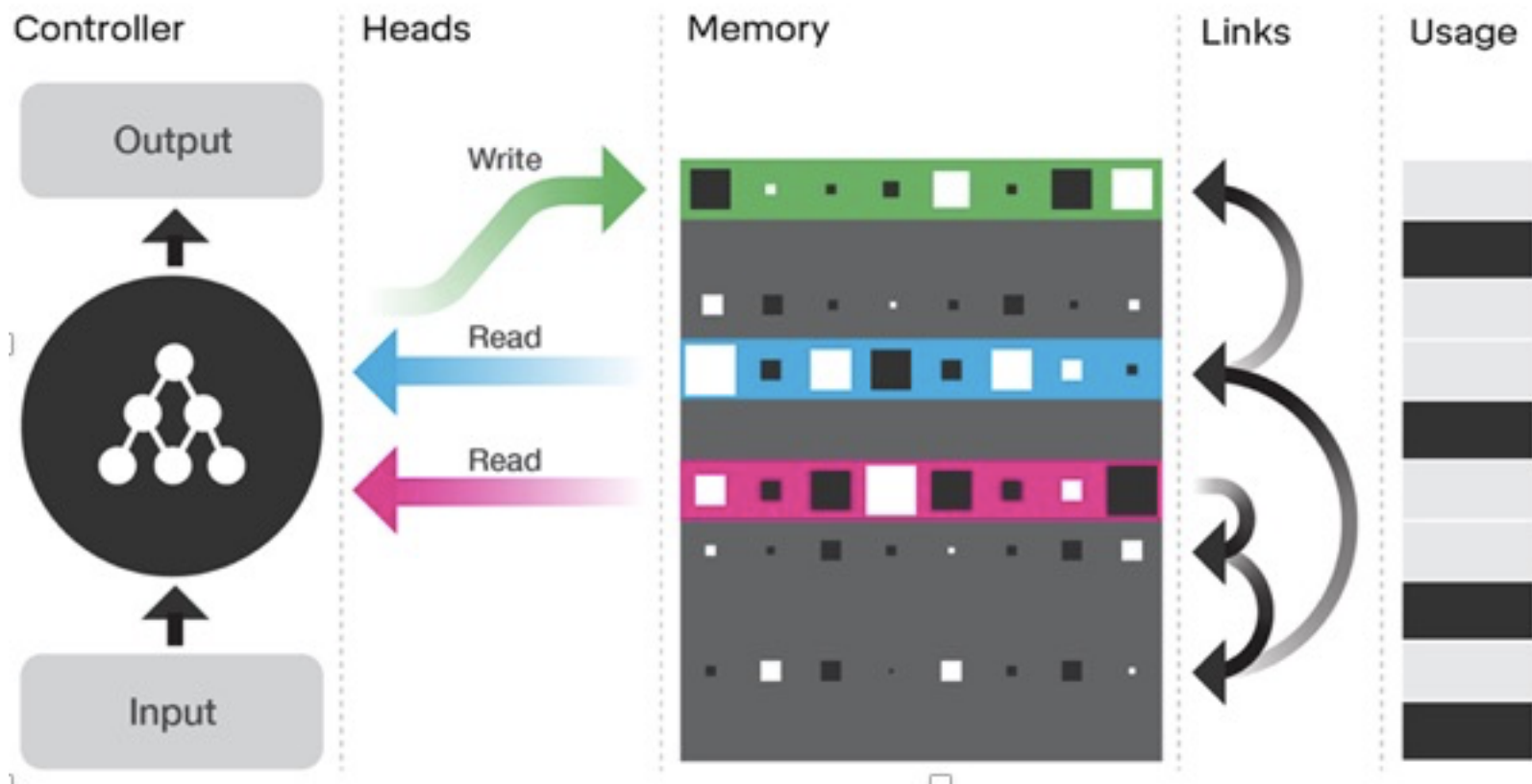
最近のブレークスルーは、ニューラルネットワークは感覚の処理、シーケンスの学習と強化学習には、極めて熟練していることを示しているのだが、認知科学者や神経科学者は、ニューラルネットワークには、変数やデータ構造を表現する能力と外部からの干渉なしに長い期間にわたってデータを格納する能力では限界があると論じてきた。

我々は、ニューラルネットワークに外部メモリへの読み書きのアクセスを提供することで、ニューラルコンピューティングとコンピュータによる処理の利点を組み合わせることを目指している。

メモリへのアクセスは、記憶間の干渉を最小限に抑える、かつ、長期間の保存を可能にする為に、狭いところに集中して行われる。システム全体は微分可能であり、それ故、勾配降下法でエンドツーエンドで訓練することができ、ネットワークは目標指向の方法でメモリを操作し、編成する方法を学習できる。

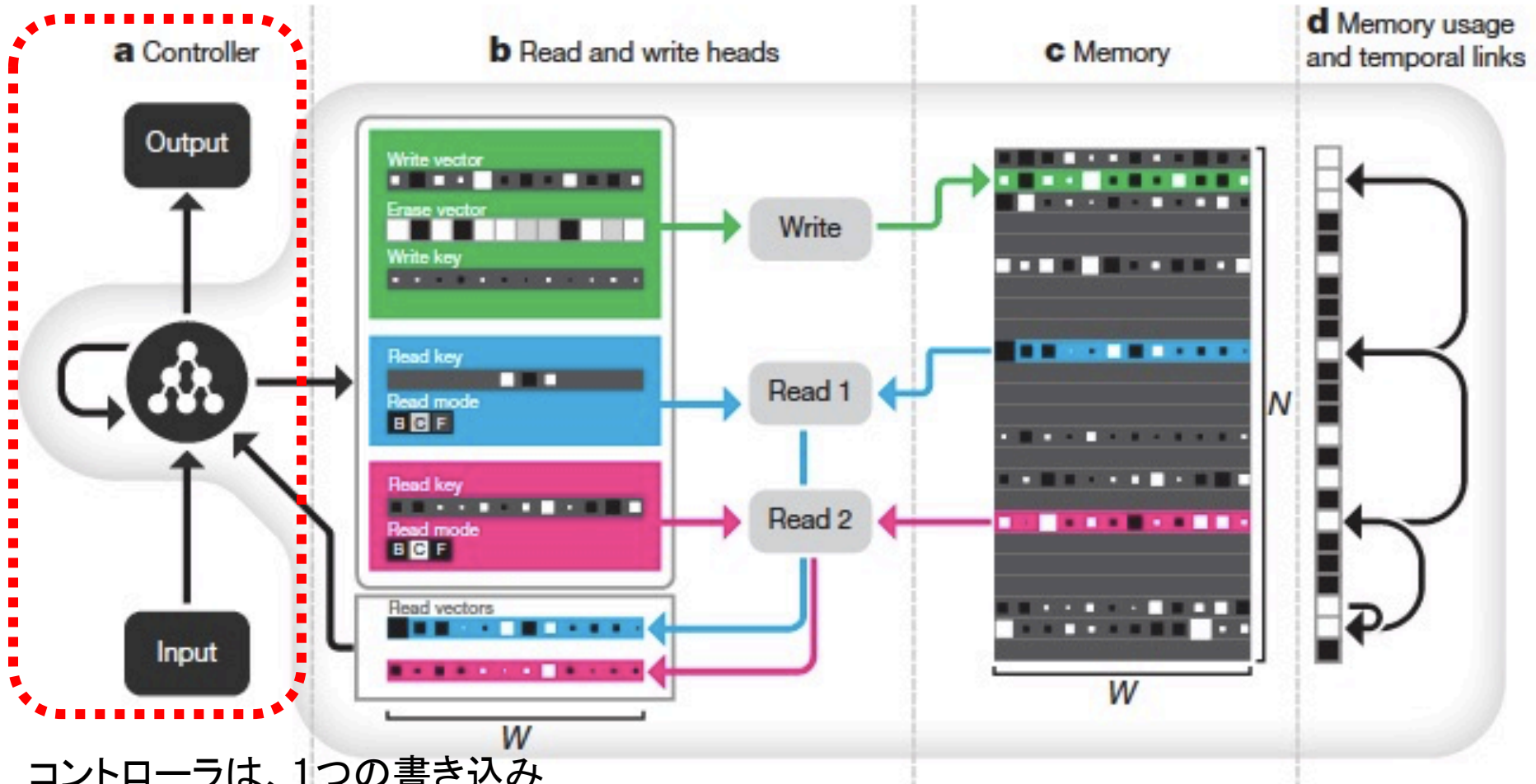
ニューラルネットワークに外部メモリへの読み書きのアクセスを提供する

DNCシステムの概観



Controller

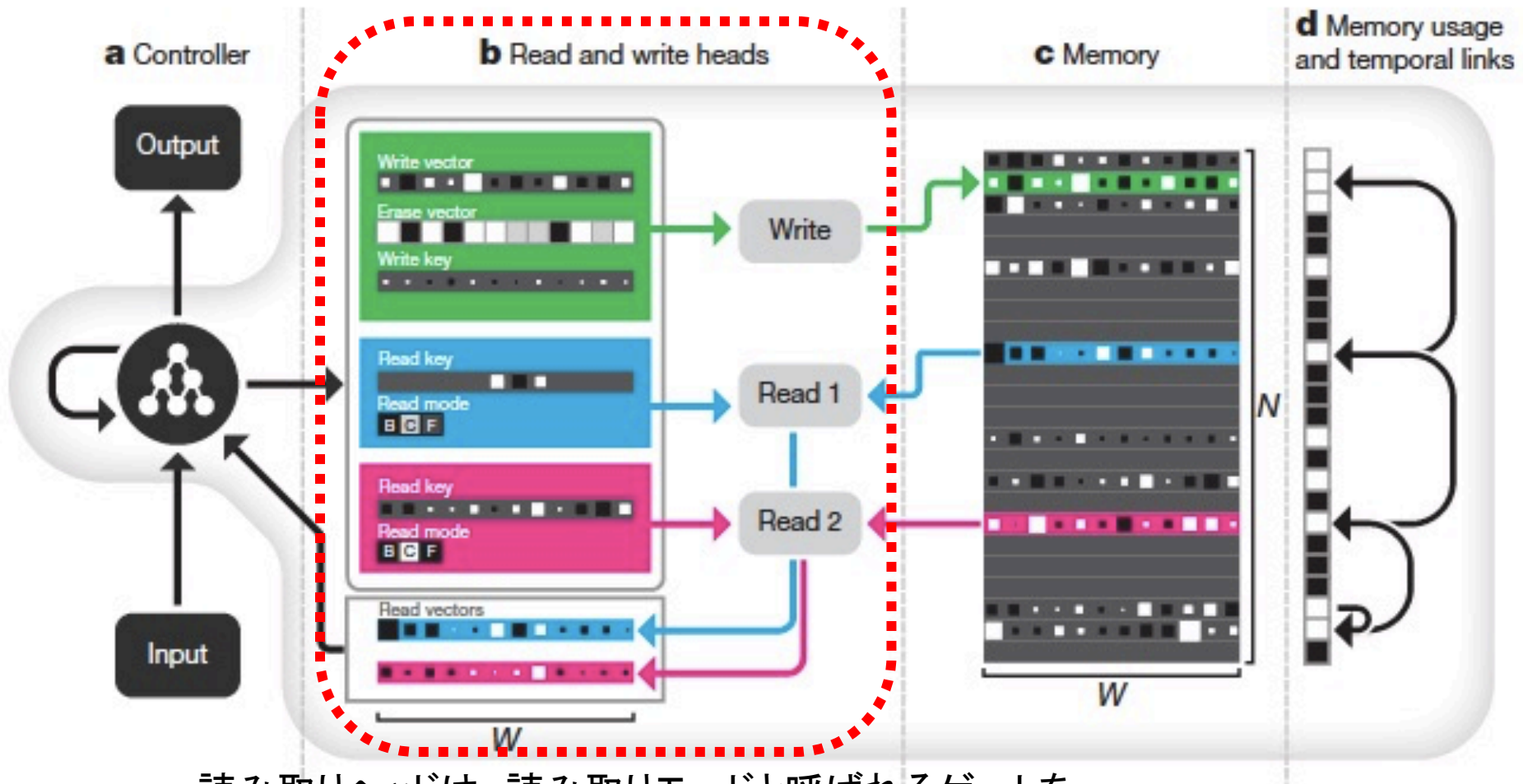
リカレント・コントローラ・ネットワークは、外部のデータソースから入力を受け取り、出力を生成する。



コントローラは、1つの書き込みヘッド(緑)と複数の読み取りヘッド(この場合は2つの青とピンク)をパラメータ化するベクトルも出力する。

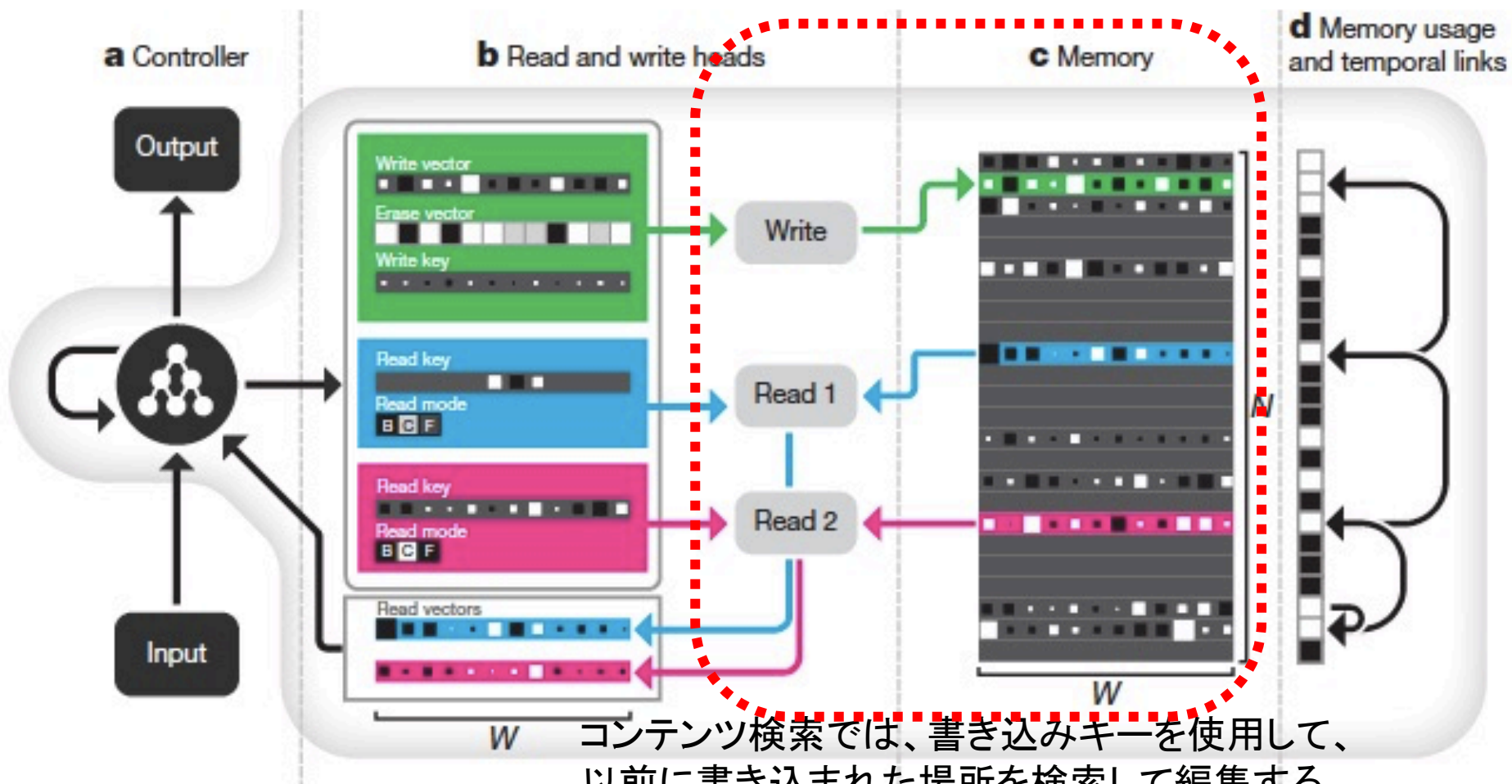
Head

書き込みヘッドは、 $N \times W$ メモリ行列を編集するために使用される書き込みベクトルおよび消去ベクトルを定義する。



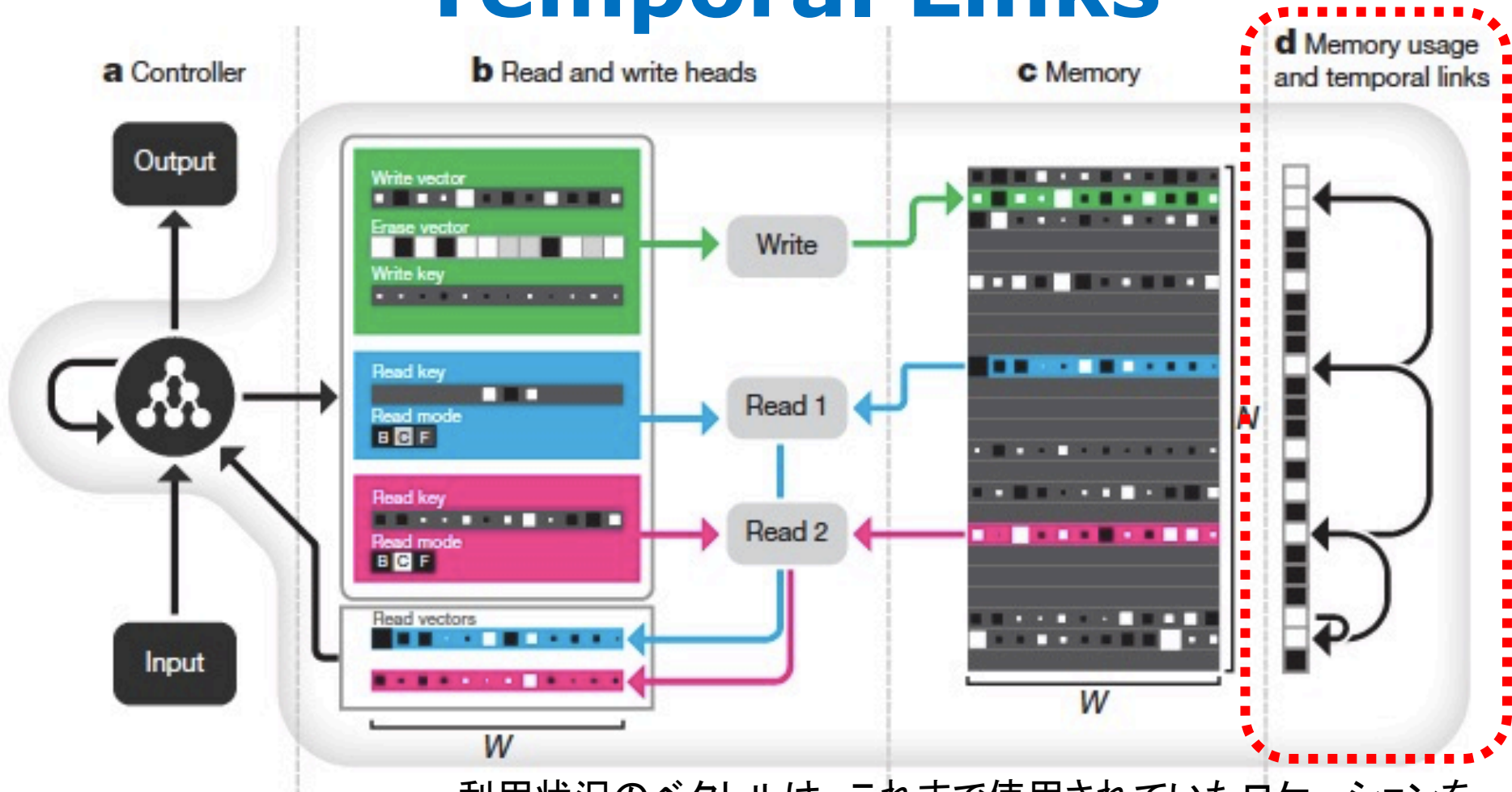
読み取りヘッドは、読み取りモードと呼ばれるゲートを使用して、読み取りキー（'C'）を使用したコンテンツ検索と、書かれた順に前方（'F'）または後方（'B'）へ進む読み出しとを切り替える。

Content Lookup



コンテンツ検索では、書き込みキーを使用して、以前に書き込まれた場所を検索して編集する。書き込みキーは、メモリ行列内の行すなわちロケーションにわたって書き込み動作を**選択的に集中させる**重み付けの定義に、影響を与えることができる。

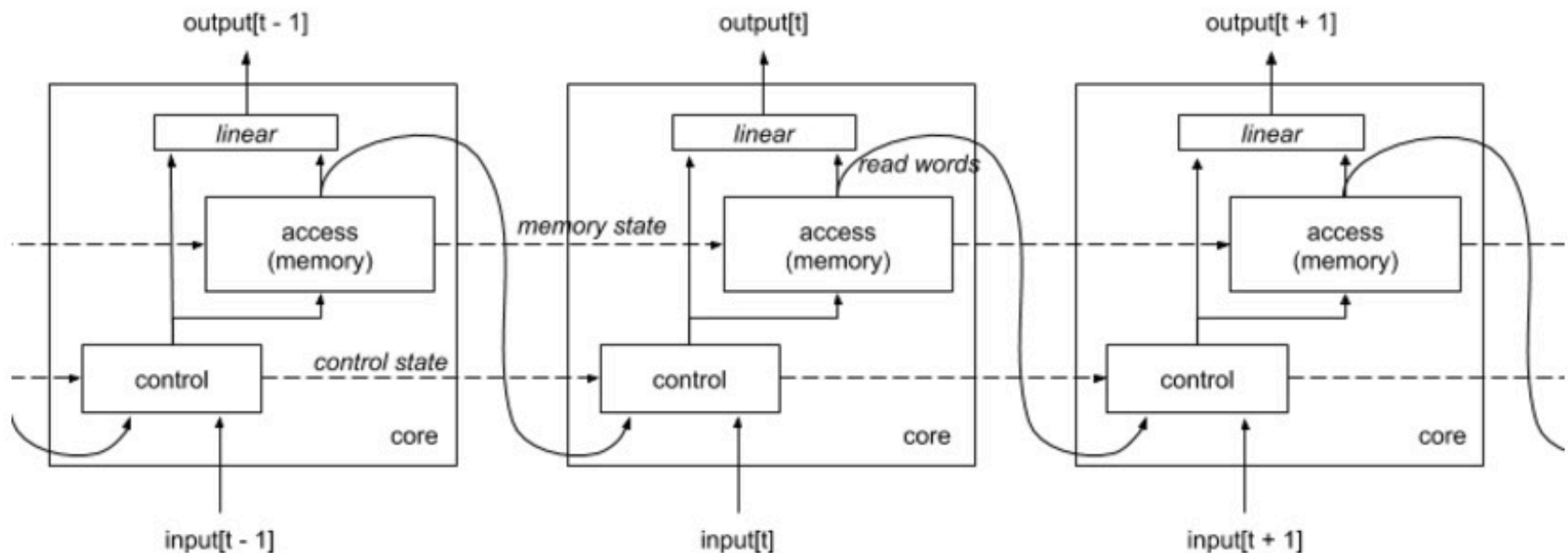
Memory Usage and Temporal Links



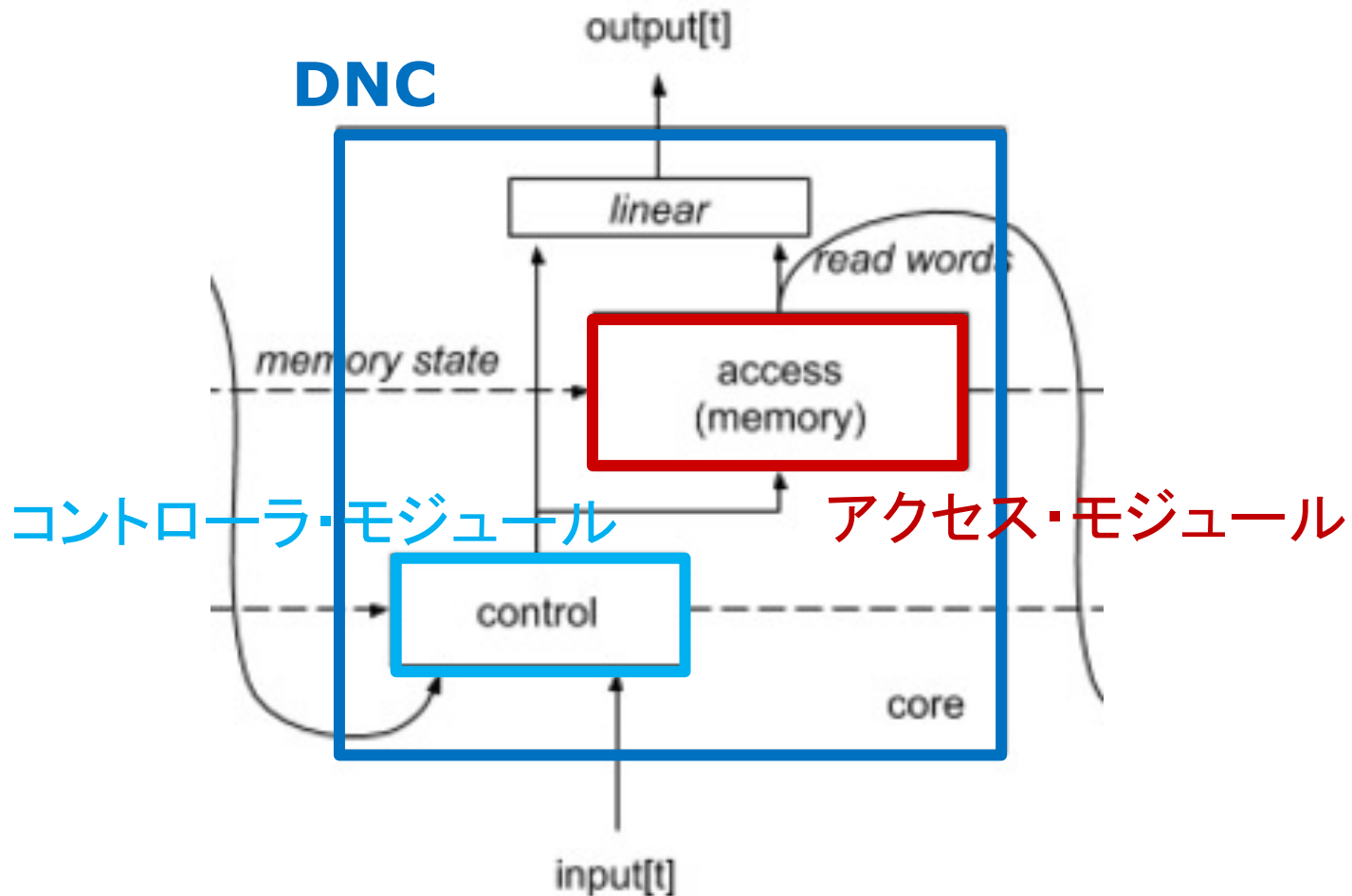
利用状況のベクトルは、これまで使用されていたロケーションを記録し、時間的リンク行列は、ロケーションに書き込まれた順序を記録する。ここでは、ロケーションの順序は、矢印で表されている。

DNCはリカレント・ニューラルネットワーク

- DNCはリカレント・ニューラルネットワークである。
各タイムステップでは、現在のメモリ内容（およびメモリ使用量などの補助的な情報）からなる状態を持ち、時刻 t の入力を時刻 t の出力にマップする。

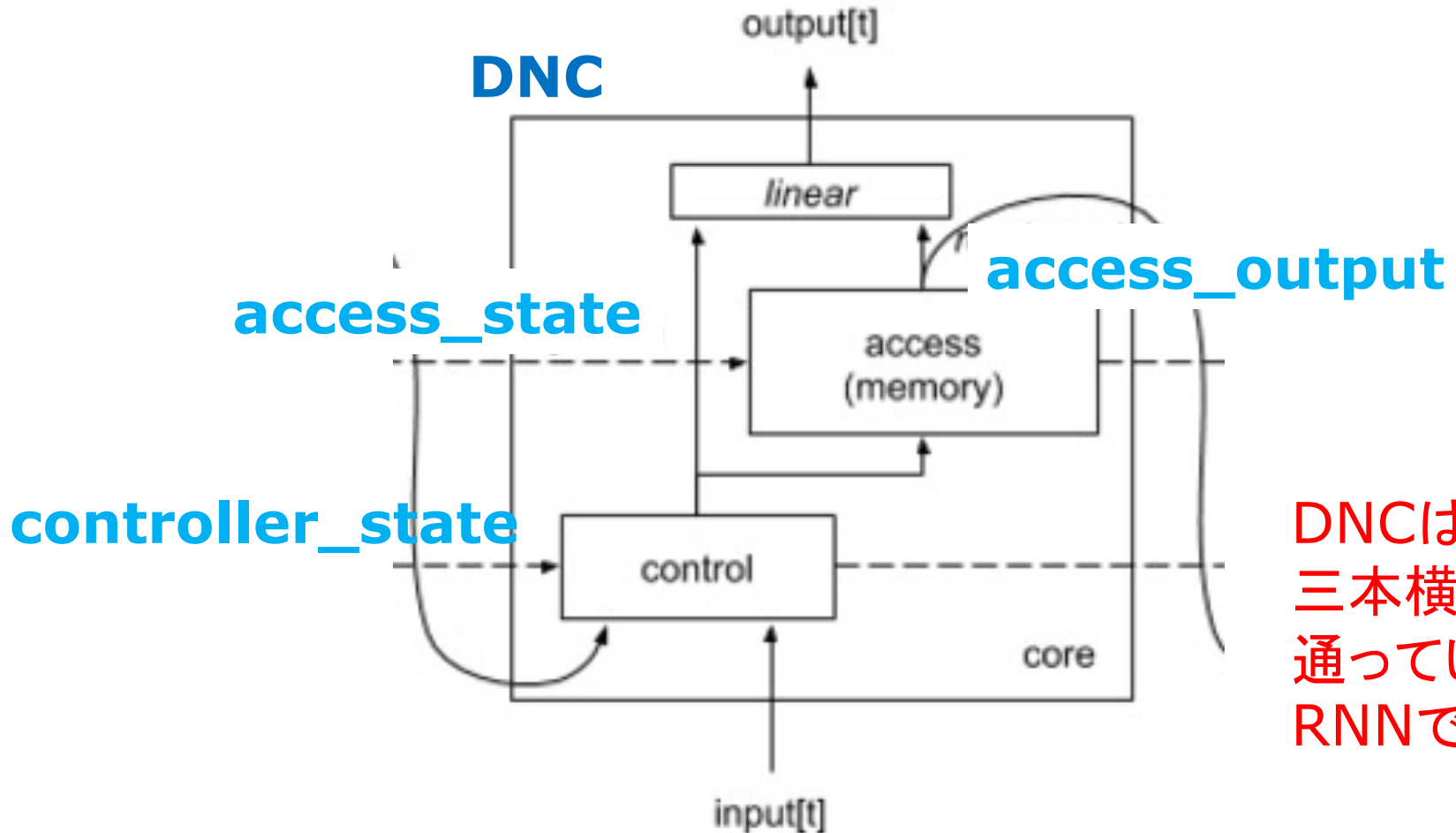


DNCは、アクセス・モジュールとコントローラ・モジュールをラップしたものの



<https://github.com/deepmind/dnc>

DNC の3つのState



DNCは、
三本横串が
通っている
RNNである

DNCが取り組んだ 三つの課題

- Synthetic question answering experiments
- Graph experiments
- Block puzzle experiments

Synthetic question answering experiments

John is in the playground.
Bob is in the office.
Where is John? **A:playground**

John is in the playground.
Bob is in the office.
John picked up the football.
Bob went to the kitchen.
Where is the football? **A:playground**
Where was Bob before the kitchen? **A:office**

John picked up the apple.
John went to the office.
John went to the kitchen.
John dropped the apple.
Where was the apple before the kitchen? **A:office**

我々の最初の実験では、質問応答を実行するDNCの能力を調査した。

DNCと他のニューラルネットワークアーキテクチャを比較するために、bAbIデータセットを検討した。これには、テキスト推論の側面を真似るように設計された20種類の総合的な質問が含まれている。

データセットは、短い「ストーリー」スニペットとそれに続くストーリーから推測できる答えの質問から構成される。

例えば、「ジョンはグラウンドにいます。ジョンはフットボールをピックアップしました」というストーリーに続いて「フットボールはどこにありますか？」という質問に、「グラウンド」と答えには、システムは2つの裏付けの事実を組み合わせる必要がある。「羊はオオカミを恐れています。」「ガートルードは羊です。」「ネズミはネコを恐れています。」「ガートルードは何を恐れているのですか？」(答え、「オオカミ」)では、基本的な演繹能力(および引っ掛けへの耐性)をテストする。

baBiについては、<https://research.fb.com/downloads/babi/> あるいは、<https://github.com/facebook/bAbI-tasks> を参照されたい。Appendixに、20タイプ全てについて、サンプルの一部を載せておいた。

bAbI best results

Task	bAbI Best Results						
	LSTM (Joint)	NTM (Joint)	DNC1 (Joint)	DNC2 (Joint)	MemN2N (Joint) ²¹	MemN2N (Single) ²¹	DMN (Single) ²⁰
1: 1 supporting fact	24.5	31.5	0.0	0.0	0.0	0.0	0.0
2: 2 supporting facts	53.2	54.5	1.3	0.4	1.0	0.3	1.8
3: 3 supporting facts	48.3	43.9	2.4	1.8	6.8	2.1	4.8
4: 2 argument rels.	0.4	0.0	0.0	0.0	0.0	0.0	0.0
5: 3 argument rels.	3.5	0.8	0.5	0.8	6.1	0.8	0.7
6: yes/no questions	11.5	17.1	0.0	0.0	0.1	0.1	0.0
7: counting	15.0	17.8	0.2	0.6	6.6	2.0	3.1
8: lists/sets	16.5	13.8	0.1	0.3	2.7	0.9	3.5
9: simple negation	10.5	16.4	0.0	0.2	0.0	0.3	0.0
10: indefinite knowl.	22.9	16.6	0.2	0.2	0.5	0.0	0.0
11: basic coreference	6.1	15.2	0.0	0.0	0.0	0.1	0.1
12: conjunction	3.8	8.9	0.1	0.0	0.1	0.0	0.0
13: compound coref.	0.5	7.4	0.0	0.1	0.0	0.0	0.2
14: time reasoning	55.3	24.2	0.3	0.4	0.0	0.1	0.0
15: basic deduction	44.7	47.0	0.0	0.0	0.2	0.0	0.0
16: basic induction	52.6	53.6	52.4	55.1	0.2	51.8	0.6
17: positional reas.	39.2	25.5	24.1	12.0	41.8	18.6	40.4
18: size reasoning	4.8	2.2	4.0	0.8	8.0	5.3	4.7
19: path finding	89.5	4.3	0.1	3.9	75.7	2.3	65.5
20: agent motiv.	1.3	1.5	0.0	0.0	0.0	0.0	0.0
Mean Err. (%)	25.2	20.1	4.3	3.8	7.5	4.2	6.4
Failed (err. > 5%)	15	16	2	2	6	3	2

Google

Facebook

実験結果をどう見るのか？ baBi task

- baBi task については、“best result” と “mean result” の両方を見た方がいいように思う。「苦手な問題」があるように思う。これは、おそらくFacebookのMemory Network モデルでも同様に見える。
- 結果的には、“best result” でも「帰納」と「位置推論」で失敗しているが、根拠から演繹的に推論するTask2, Task3も、“mean result” を見る限り、安定していないように見える。
- 比較するには、いくつかの前提が必要なのだが、論理的推論能力に関していえば、最も最初期の自動証明プログラムであるSimonの“LOGIC THEORIST”（「プリンキピア・マテマティカ」の最初の52の定理のうち38の定理を証明できた）の能力を大きく超えてはいないようにも思う。

実験結果をどう見るのか？ baBi task

- 具体的な文が直接に表すものを理解するのと、そこから論理的構造を抽象して、それに基づいて推論するのは、違うレベルの「知能」の働きなのだ。
- おそらく、そこには超えなければいけない課題が何かある。

実験結果をどう見るのか？ Graph task

- Graph taskについては、もう一つの「方法論的」と言ってもいい、悩ましい問題がある。それは、コンピュータは、グラフの問題を別の方法でも解くことができるということである。
- 「最小経路」の問題は、Dijkstraによるよく知られたアルゴリズムがある。スマホの経路検索でも普通に使われているものだ。DNCは、「総当たり」法を発見し、それを試そうとしているように見える。DNCは、Dijkstraの方法を発見できるのだろうか？ 多分、無理だと思う。
- もっと大事なことは、グラフで表現された知識を大規模なシステムで処理するというのは、ディープ・ラーニングでもルール・ベースでもないのだが、広い意味での、そして最も身近な、現代の「人工知能」技術の、すでに根幹であるということだ。
- 2014年2月のマルレク「大規模グラフデータ処理」
<https://goo.gl/n8WuRb> を参照されたい。

実験結果をどう見るのか？

- Graph Taskのサンプルは、僕には、デジャブだった。
これは、Prologの練習問題だ。このレベルのデータ量なら、すぐ解けるはずだ。
- Block Taskは、やはり、WinogradのSHRDLUとの比較が必要だと思う。強化学習を使うというのだが、baBi Task 17の、位置の推論が苦手というのは、こうした課題には、大きなハンディだと思う。

DNCはスケールできるか？

- Gravesはいう。
「実世界のデータに取り組むためには、数千または数百万のロケーションにスケールアップする必要がある。その時点で、メモリはコントローラの重みに含まれる情報より多くの情報を格納できる。」
- ただ、DNCのアーキテクチャーを、そうした規模に拡張できるのだろうか？
- 実際に、DNCが三つの実験に使ったメモリーのロケーションの数は、最大256である。

人間と機械翻訳のギャップに橋をかける Googleのニューラル機械翻訳システム

"Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation"

Yonghui Wu et al.

<https://goo.gl/YqIAAW>

2016年



Googleの多言語ニューラル機械翻訳

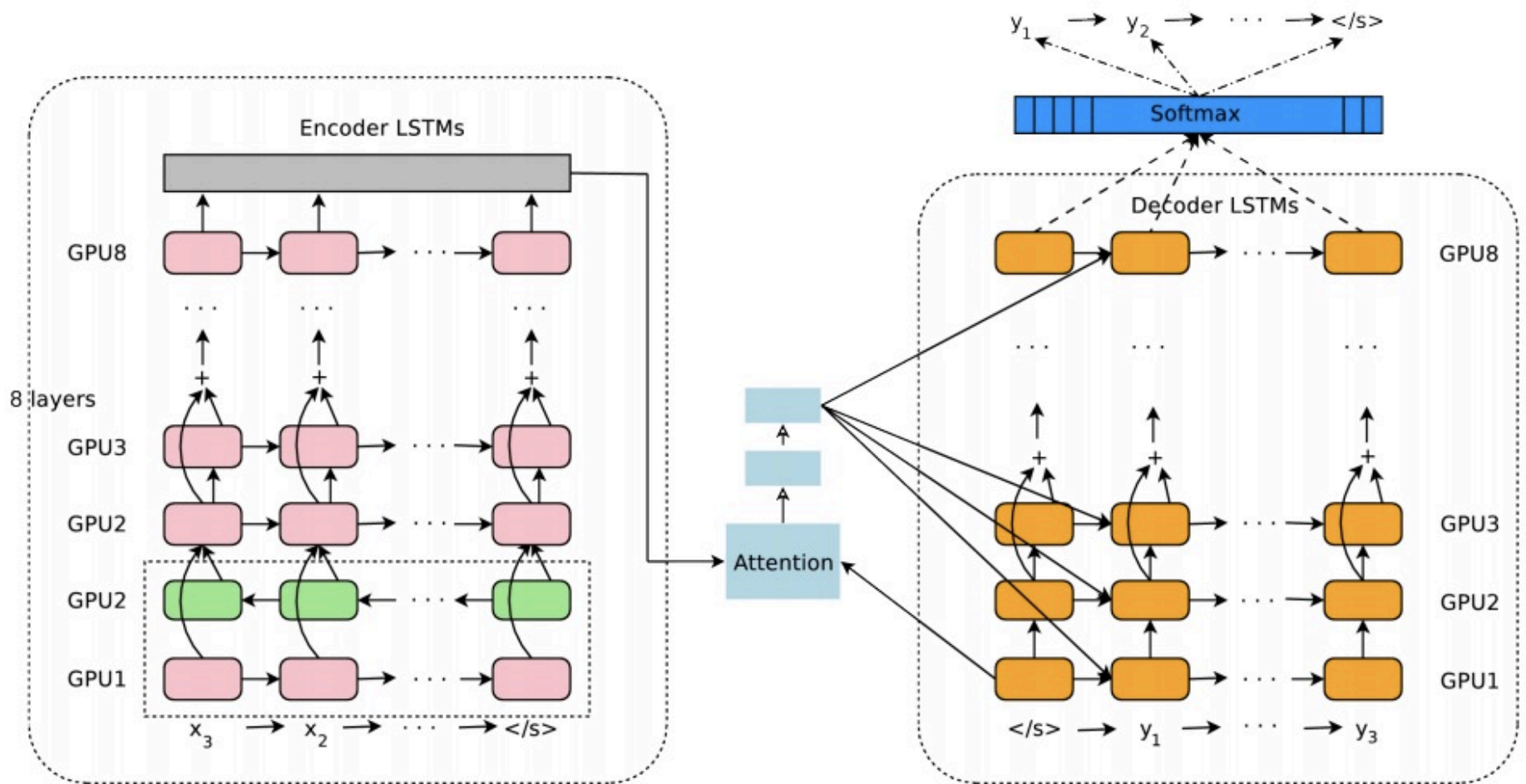
"Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation"

Melvin Johnson et al.
<https://goo.gl/islUXa>

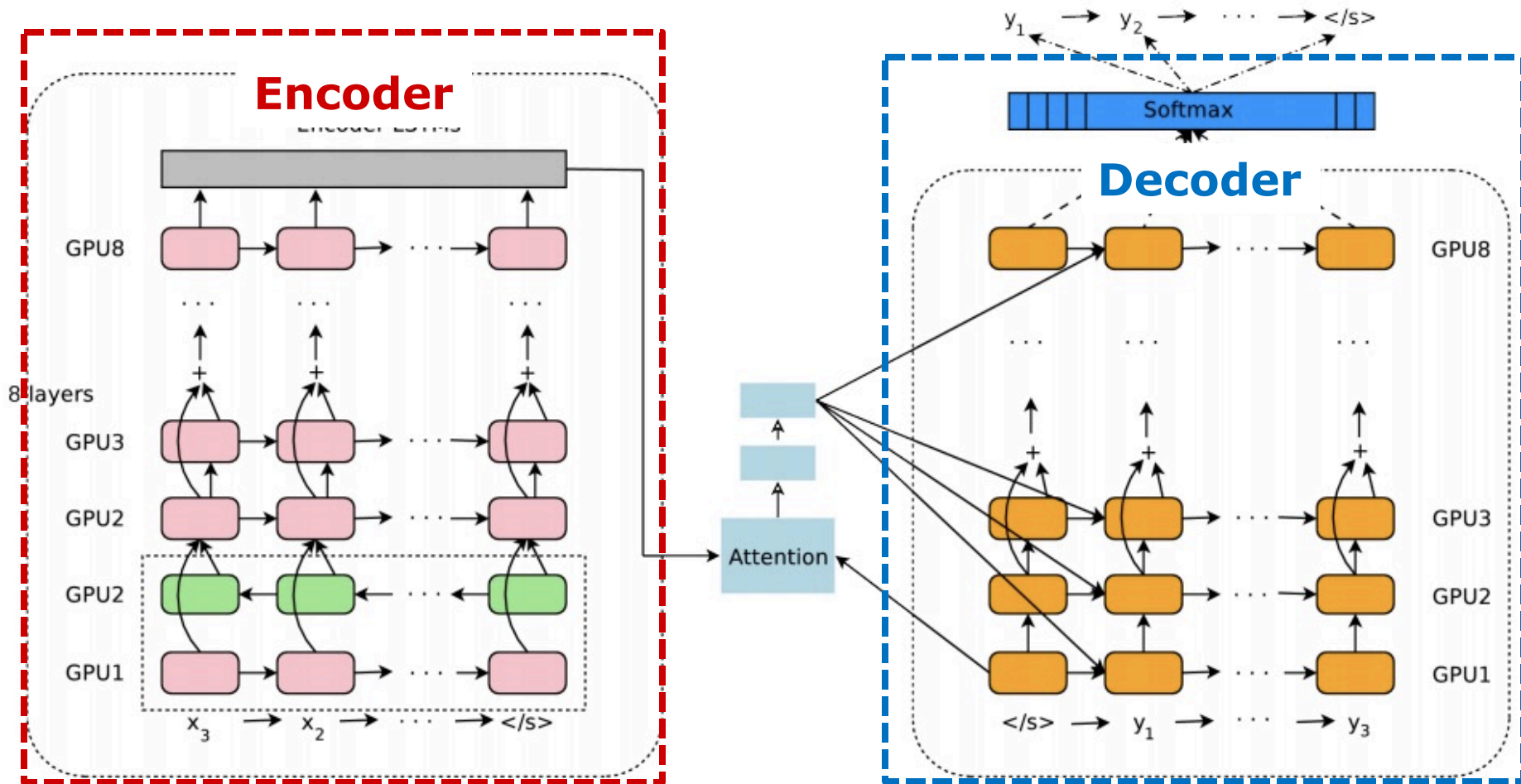
2016年



GNMTのアーキテクチャの概念図

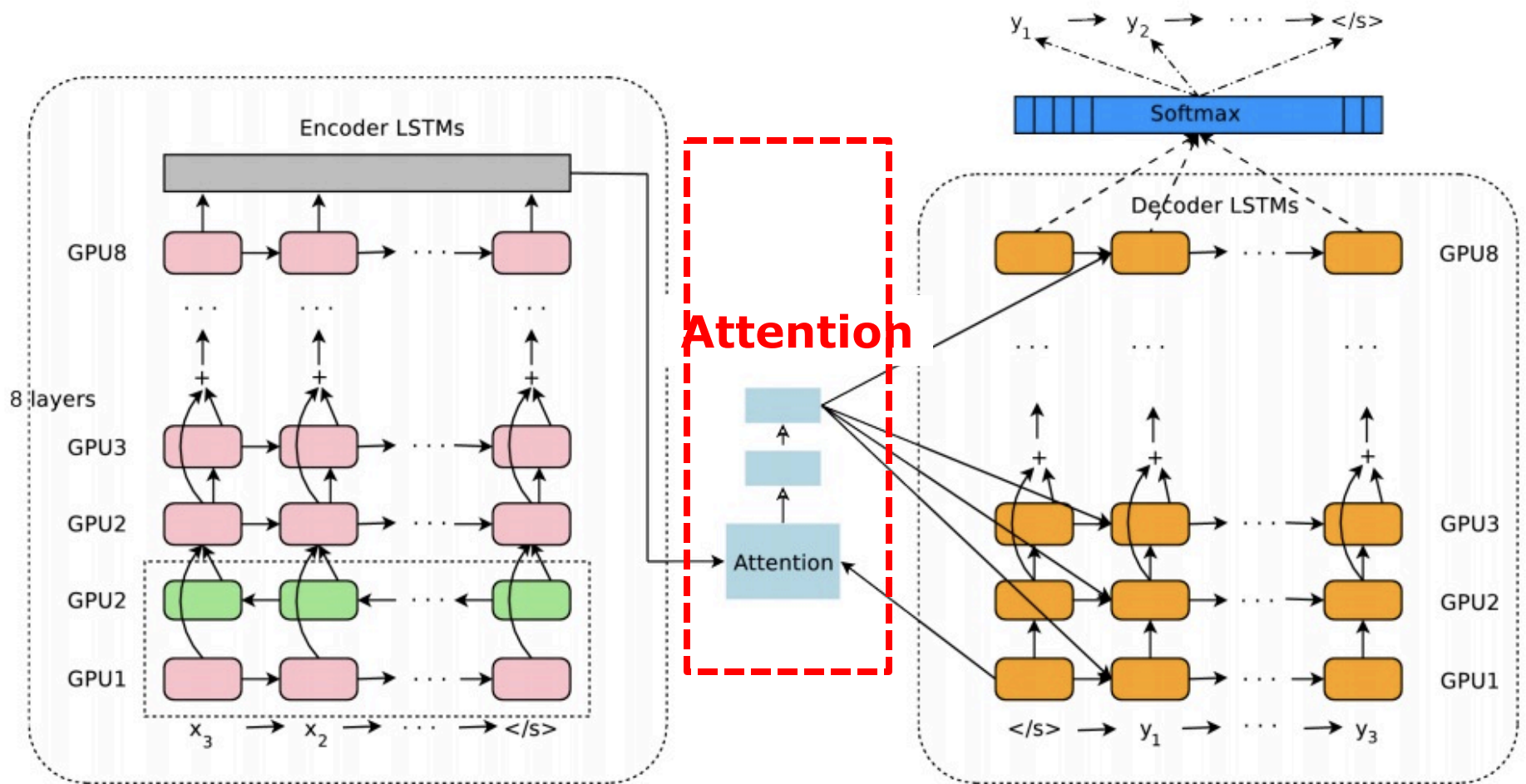


Encoder / Decoder



左側に、LSTMを8段重ねにした Encoder LSTMがあり、右側には、同じくLSTMを8段重ねにした Decoder LSTMがある。

Attention Mechanism



EncoderとDecoderの中間に、Attentionと記された領域がある。ここからの出力Attention Contextは、Decoderのすべてのノードに供給されている。

ビジュアルな分析 インターリンガの存在

言語にかかわらず、ネットワークは、同じ意味を持つ文章が同じような方法で表現される何らかの共有表現を学習しているのか？ この質問に対しては、そうだという。これも、画期的。モデルは、訓練された言語ペアを扱うのと同じ方法で、ゼロ・ショット翻訳を実行しているのか？ これについては、まだよくわからないという。

実験結果が提起する問題

- モデルを複数の言語にまたがってトレーニングすることで、個々の言語レベルでのパフォーマンスが向上し、ゼロ・ショット翻訳が有効になることがわかるということが、この論文の結論なのだが、この結果は、これらのタスクがモデル内でどのように処理されるかについて、次のような問題を提起する。
 1. 言語にかかわらず、ネットワークは、同じ意味を持つ文章が同じような方法で表現される何らかの共有表現を学習しているのか？
 2. モデルは、訓練された言語ペアを扱うのと同じ方法で、ゼロ・ショット翻訳を実行しているのか？
-

ビジュアル化による分析

- ネットワークによって使用される表現を研究する1つの方法は、翻訳中のネットワークの活性化の状態を見ることである。
- 研究の出発点は、アテンション・ベクトルのセット、すなわち、エンコーダとデコーダのネットワークを接続するレイヤ内の活性化の状況を見ることだ。
- 単一の文の翻訳は、一連のアテンション・ベクトルを生じさせる。この文脈において、共有表現に関する元々の問題は、異なる文のベクトルのシーケンスがどのように関連しているかを調べることで研究することができる。例えば、次のように。
- ソースまたはターゲット言語に応じて、文章がクラスタリングされているのか？
- あるいは、言語にかかわらず、同様の意味を持つ文章がクラスタリングされるのか？

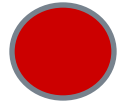
インターリンガの証拠

- いくつかの訓練されたネットワークは、実際に共有表現の強いビジュアルな証拠を示す。
- たとえば、以下の図2は、英語 <-> JapaneseとEnglish <-> Koreanでトレーニングされた多対多モデルから作成されたものである。モデルの実際の動作を視覚化するために、意味論的に同一の言語間フレーズの74個のトリプルからなる小さなコーパスから始めた。
- つまり、それぞれのトリプルには、同じ基本的な意味を持つ英語、日本語、韓国語のフレーズが含まれている。これらのトリプルをコンパイルするために、私たちは日本語と韓国語の翻訳と対になった、英語の文章のための正しい(Ground Truth)データベースを検索した。

英語



三つの言語で
同じ意味を持つ文を
一つのトリプルに
まとめる。



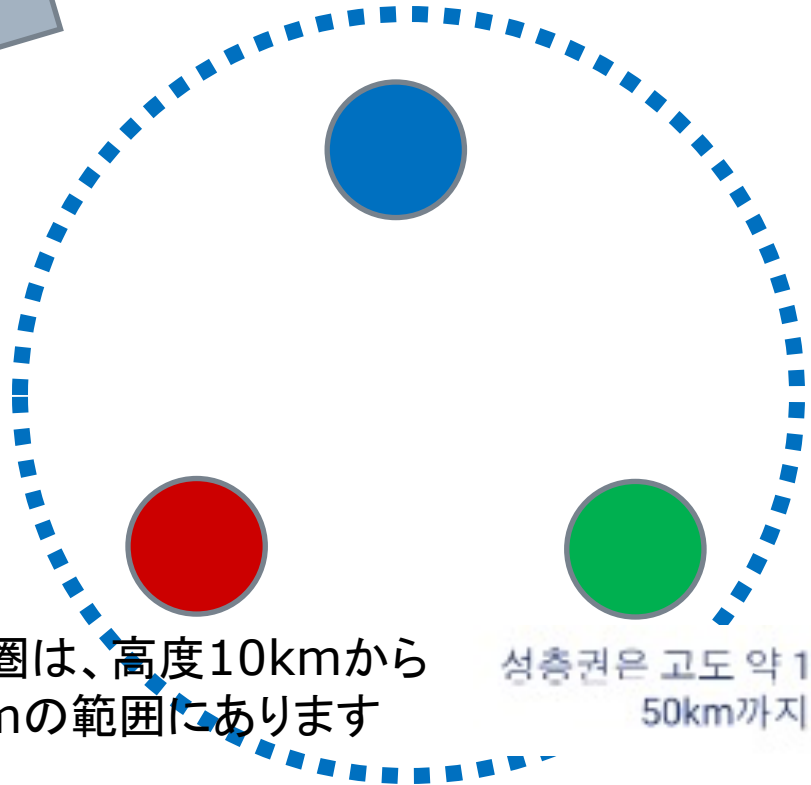
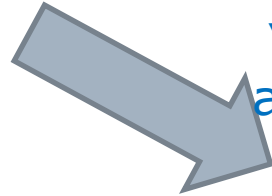
日本語



韓国語

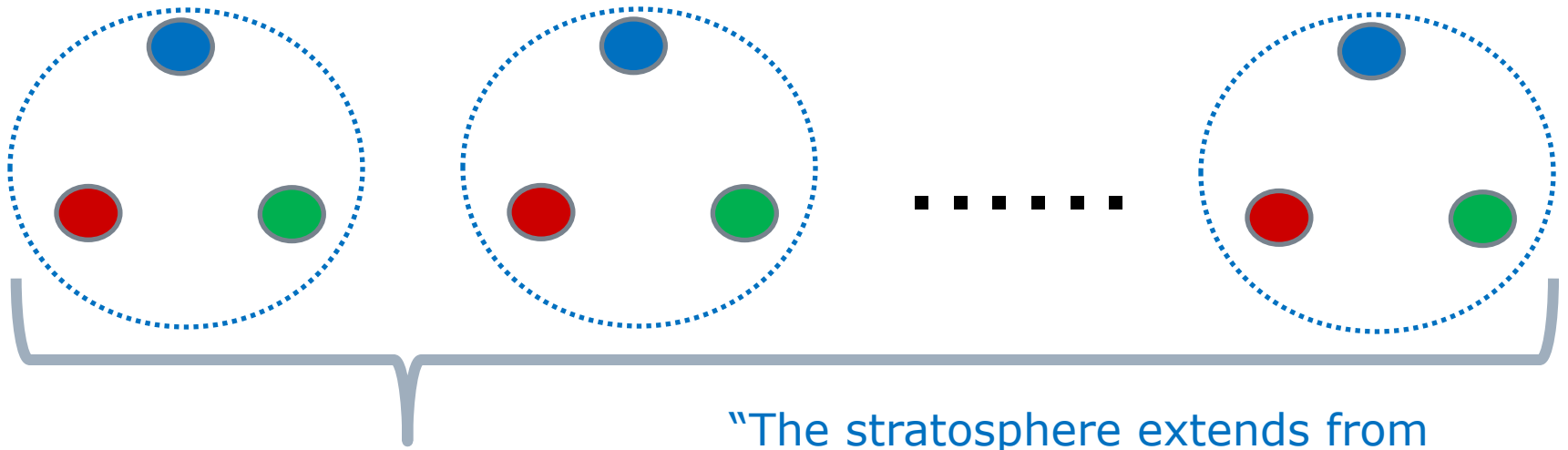
例えば

“The stratosphere extends from
about 10km to about 50km in altitude.”



成層圏は、高度10kmから
50kmの範囲にあります

성층권은 고도 약 10km부터 약
50km까지 확장됩니다.

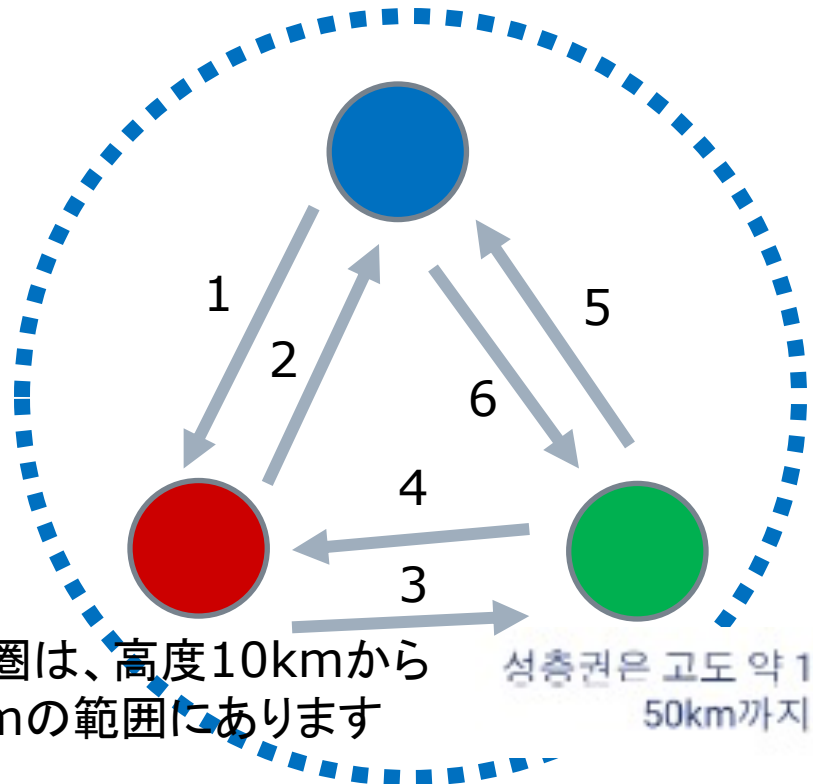


74個のトリプルからなるコーパスを準備する。

各トリプルの各センテンスを他の2つの言語に翻訳する。一つのトリプルについて、6つの翻訳が生まれる。

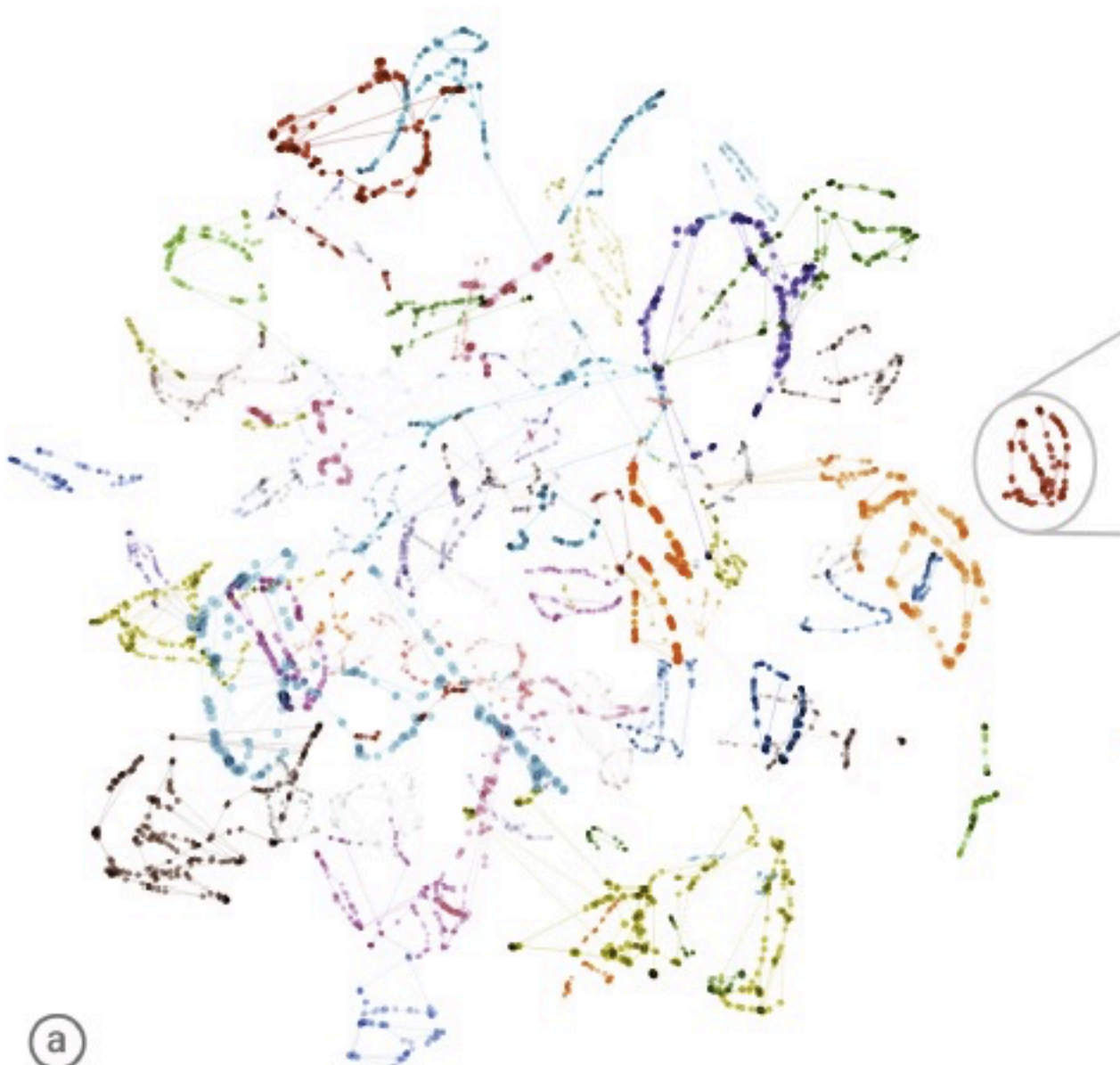
このコーパスでは、 $74 \times 6 = 444$ の翻訳が生まれる。

“The stratosphere extends from about 10km to about 50km in altitude.”



成層圏は、高度10kmから50kmの範囲にあります

성층권은 고도 약 10km부터 약 50km까지 확장됩니다.



a

言語にかかわらず、同様の意味を持つ文章がクラスタリングされている。

この 444 の翻訳を行うのに、システムは 9,978 ステップ要した。

この時、発せられた 9,978 個のアテンションベクトルを、t-SNE を用いて二次元に投射したのが左の図。

同じトリプル内の文の翻訳で発せられるアテンション・ベクトルは、同じ色で表している。

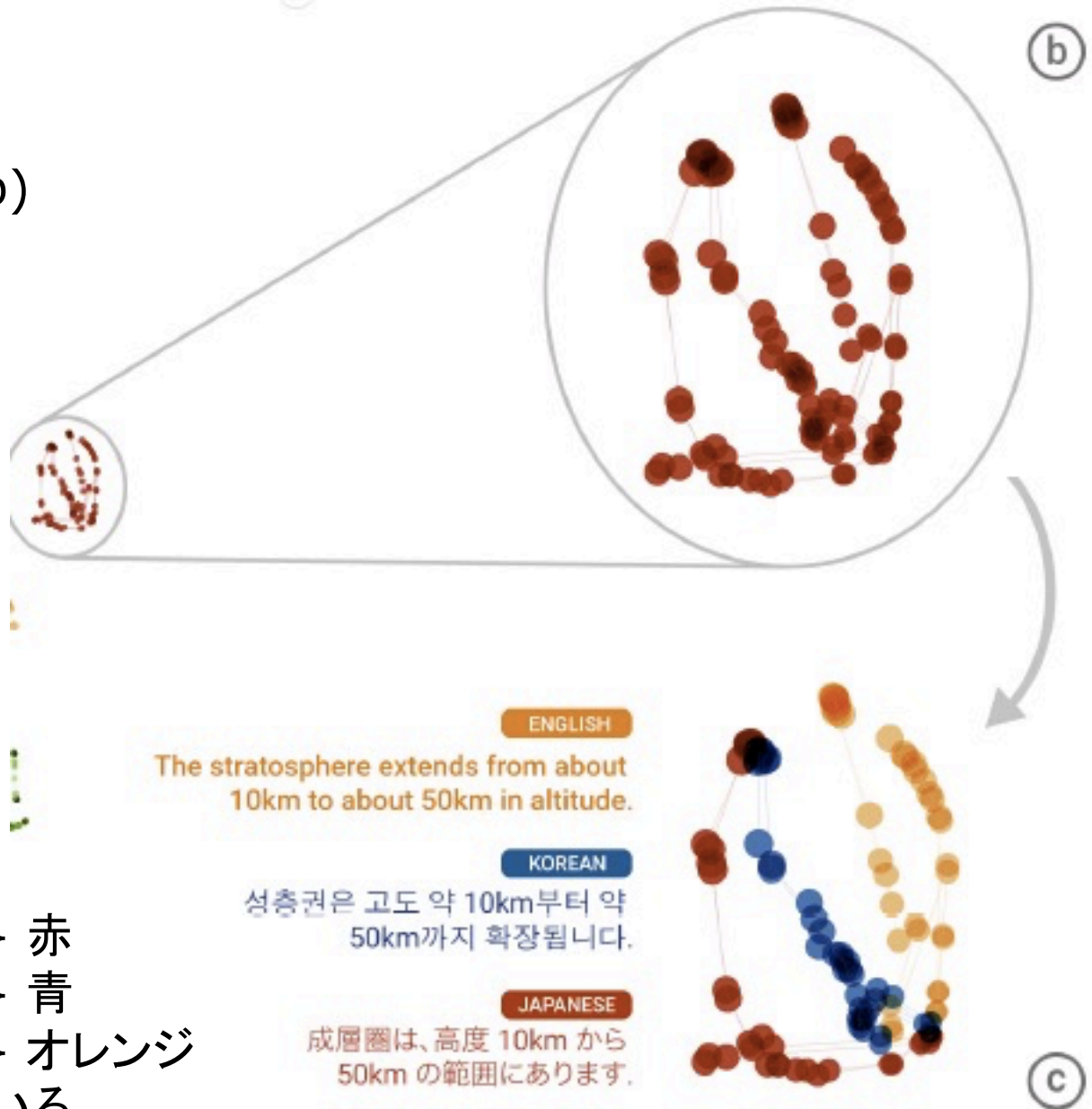
同じ色が、近くに集まってクラスターを形成しているのが見て取れる。
(?)

この同じ色のクラスター(b)
は、元の同じ意味を持つ
トリプルから生まれたもの
である。

同一の文から発せられる
アテンション・ベクトルは、
線で結ばれている。

(c)は、翻訳ソースが同じ
言語を同じ色で塗り分け
たもの。

日本語->{英語,韓国語} 赤
韓国語->{英語,日本語} 青
英語->{日本語,韓国語} オレンジ
は、クラスターを形成している。



パラレル・コーパスの現在

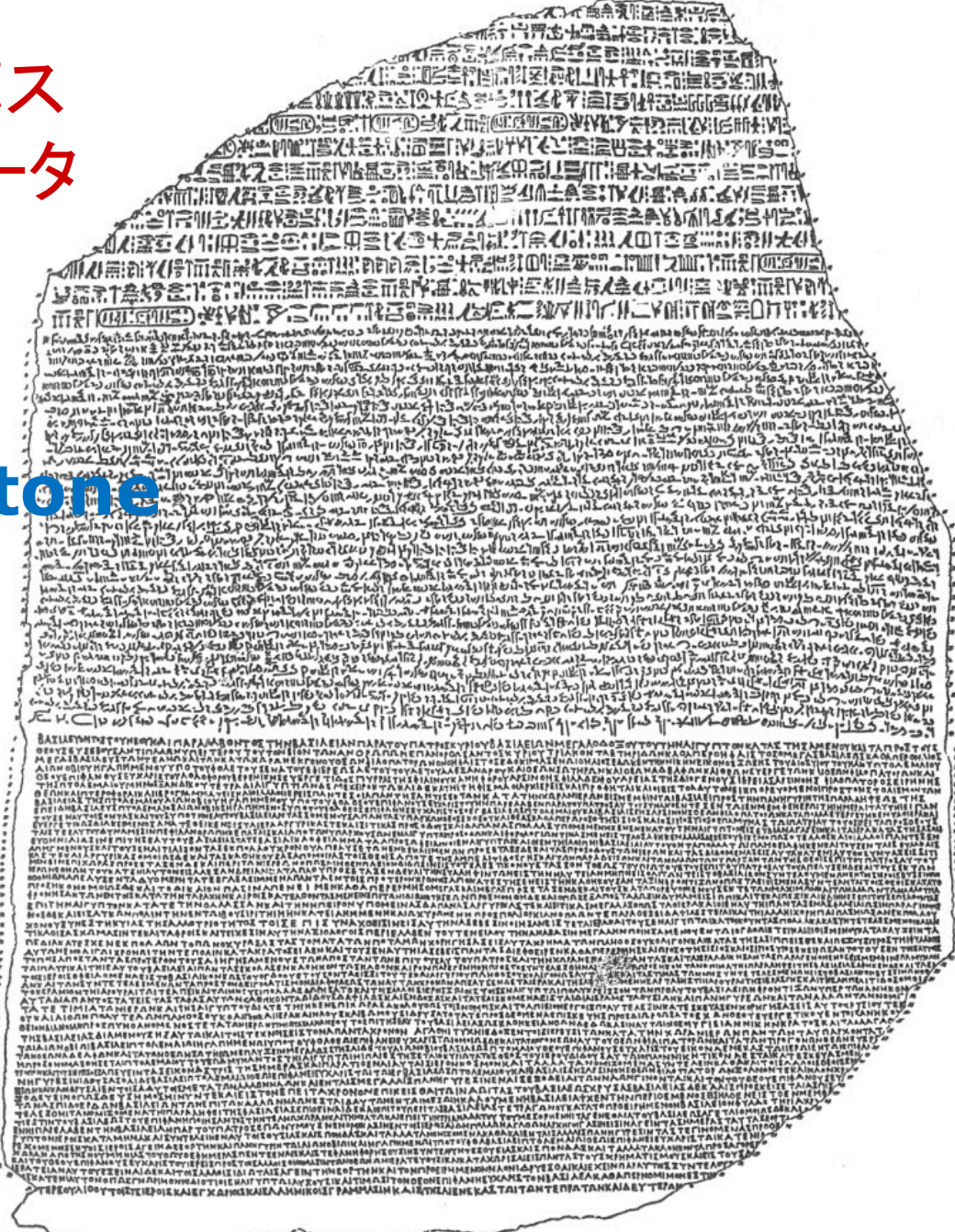
ここでは、少し視点を変えて、SMTにとってもNMTにとっても本質的に重要な、コーパスの現在を見てみよう。驚くほどの大規模化が進んでいる。

「勝つのは、最良のアルゴリズムを持っている人ではなく、もっとも多くのデータを持っている人である。」という主張が、強い説得力を持つことは疑いない。ただ一点、「最良のアルゴリズム」の持つ、破壊的なインパクトを軽視していることを除けば。

最古のコーパス パラレル・データ

Rosetta Stone BC 196年

1822年
シャンポリオンが解読



神聖文字

民衆文字

ギリシャ
文字

最古の平行・データ

ベヒストウン碑文 楔形文字の解読

- コーパス: ベヒストウン碑文 BC522~
 (エラム語、古代ペルシア語、アッカド語
 の三つの言語で書かれている)
 解読者: ローリンソンとヒンクス
 1846-1851



ベヒストウン碑文のパピロン

ペルシャ楔形文字

	BA	A	BA	I	RU	U

エラム語表記

	BA	PI	LA
	12040	1227	12167

新アッシリア字形

アッカド語表記

	DIN	TIR	KI
	12077	12301	121a0

ヒッタイト字形

コーパスの例(英仏)

English	French
<p>According to our survey, 1988 sales of mineral water and soft drinks were much higher than in 1987, reflecting the growing popularity of these products. Cola drink manufacturers in particular achieved above-average growth rates. The higher turnover was largely due to an increase in the sales volume. Employment and investment levels also climbed. Following a two-year transitional period, the new Foodstuffs Ordinance for Mineral Water came into effect on April 1, 1988. Specifically, it contains more stringent requirements regarding quality consistency and purity guarantees.</p>	<p>Quant aux eaux minérales et aux limonades, elles rencontrent toujours plus d'adeptes. En effet, notre sondage fait ressortir des ventes nettement supérieures à celles de 1987, pour les boissons à base de cola notamment. La progression des chiffres d'affaires résulte en grande partie de l'accroissement du volume des ventes. L'emploi et les investissements ont également augmenté. La nouvelle ordonnance fédérale sur les denrées alimentaires concernant entre autres les eaux minérales, entrée en vigueur le 1er avril 1988 après une période transitoire de deux ans, exige surtout une plus grande constance dans la qualité et une garantie de la pureté.</p>

Gale & Church "A Program for Aligning Sentences in Bilingual Corpora"
<http://www.aclweb.org/anthology/J93-1004.pdf>から。

コーパスの例(英仏)

English	French
According to our survey, 1988 sales of mineral water and soft drinks were much higher than in 1987, reflecting the growing popularity of these products. Cola drink manufacturers in particular achieved above-average growth rates.	Quant aux eaux minérales et aux limonades, elles rencontrent toujours plus d'adeptes. En effet, notre sondage fait ressortir des ventes nettement supérieures à celles de 1987, pour les boissons à base de cola notamment.
The higher turnover was largely due to an increase in the sales volume.	La progression des chiffres d'affaires résulte en grande partie de l'accroissement du volume des ventes.
Employment and investment levels also climbed.	L'emploi et les investissements ont également augmenté.
Following a two-year transitional period, the new Foodstuffs Ordinance for Mineral Water came into effect on April 1, 1988. Specifically, it contains more stringent requirements regarding quality consistency and purity guarantees.	La nouvelle ordonnance fédérale sur les denrées alimentaires concernant entre autres les eaux minérales, entrée en vigueur le 1er avril 1988 après une période transitoire de deux ans, exige surtout une plus grande constance dans la qualité et une garantie de la pureté.

Gale & Church "A Program for Aligning Sentences in Bilingual Corpora"
<http://www.aclweb.org/anthology/J93-1004.pdf>から。

機械翻訳に利用される、現代のコーパスの規模

- WMT 14の英語(En) <-> フランス語(Fr)データセットには、3,600万の文のペアが含まれている。
- WMT 14の英語(En) <-> ドイツ語(De)データセットには、500万の文のペアが含まれている。
- Googleは、内部に、英語 <-> 日本語(Ja)、英語 <-> 韓国語(Ko)、英語 <-> スペイン語(Es)、英語 <-> ポルトガル語(Pt) 等々の多くのデータセットを持っているが、その規模は、先のWMTのデータセットより、2~3桁大きいという。
- Googleニューラル機械翻訳では、GPU100個を使って、フルトレーニングには最大1,000万ステップ、収束までには3週間かかることがあるという。

RNNの驚くべき能力

*"The Unreasonable Effectiveness
of Recurrent Neural Networks"*

Andrej Karpathy

<http://goo.gl/mNqwCv>

2015年



For $\bigoplus_{n=1, \dots, m}$ where $\mathcal{L}_{m_\bullet} = 0$, hence we can find a closed subset \mathcal{H} in \mathcal{H} and any sets \mathcal{F} on X , U is a closed immersion of S , then $U \rightarrow T$ is a separated algebraic space.

Proof. Proof of (1). It also start we get

$$S = \text{Spec}(R) = U \times_X U \times_X U$$

and the comparicoly in the fibre product covering we have to prove the lemma generated by $\coprod Z \times_U U \rightarrow V$. Consider the maps M along the set of points Sch_{fppf} and $U \rightarrow U$ is the fibre category of S in U in Section, ?? and the fact that any U affine, see Morphisms, Lemma ???. Hence we obtain a scheme S and any open subset $W \subset U$ in $Sh(G)$ such that $\text{Spec}(R') \rightarrow S$ is smooth or an

$$U = \bigcup U_i \times_S U_i$$

which has a nonzero morphism we may assume that f_i is of finite presentation over S . We claim that $\mathcal{O}_{X,x}$ is a scheme where $x, x', s'' \in S'$ such that $\mathcal{O}_{X,x'} \rightarrow \mathcal{O}'_{X',x'}$ is separated. By Algebra, Lemma ?? we can define a map of complexes $GL_{S'}(x'/S'')$ and we win. \square

To prove study we see that $\mathcal{F}|_U$ is a covering of \mathcal{X}' , and \mathcal{T}_i is an object of $\mathcal{F}_{X/S}$ for $i > 0$ and \mathcal{F}_p exists and let \mathcal{F}_i be a presheaf of \mathcal{O}_X -modules on \mathcal{C} as a \mathcal{F} -module. In particular $\mathcal{F} = U/\mathcal{F}$ we have to show that

$$\widetilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\text{Spec}(k)} \mathcal{O}_{S,s} - i_X^{-1} \mathcal{F}$$

is a unique morphism of algebraic stacks. Note that

$$\text{Arrows} = (Sch/S)_{fppf}^{opp}, (Sch/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \mapsto (U, \text{Spec}(A))$$

is an open subset of X . Thus U is affine. This is a continuous map of X is the inverse, the groupoid scheme S .

Proof. See discussion of sheaves of sets.

RNNが産み出した
数学論文モドキ

Stack Theory の教科書を
「学習」させたもの

Proof. Omitted. □

Lemma 0.1. *Let \mathcal{C} be a set of the construction.*

Let \mathcal{C} be a gerber covering. Let \mathcal{F} be a quasi-coherent sheaves of \mathcal{O} -modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

Proof. This is an algebraic space with the composition of sheaves \mathcal{F} on $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where \mathcal{G} defines an isomorphism $\mathcal{F} \rightarrow \mathcal{F}$ of \mathcal{O} -modules. □

Lemma 0.2. *This is an integer \mathcal{Z} is injective.*

Proof. See Spaces, Lemma ?? □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset X$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.*

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \rightarrow Y' \rightarrow Y \rightarrow Y \rightarrow Y' \times_X Y \rightarrow X.$$

be a morphism of algebraic spaces over S and Y .

Proof. Let X be a nonzero scheme of X . Let X be an algebraic space. Let \mathcal{F} be a quasi-coherent sheaf of \mathcal{O}_X -modules. The following are equivalent

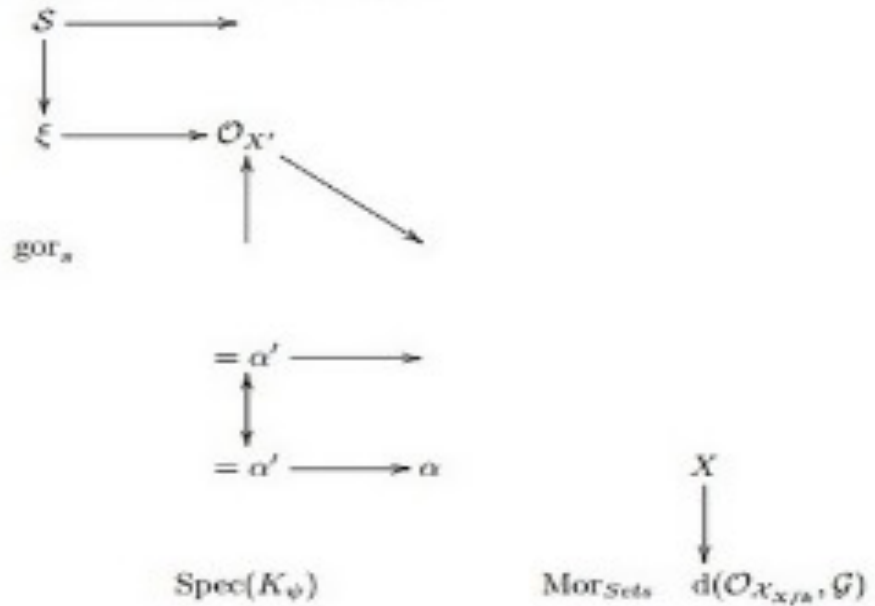
- (1) \mathcal{F} is an algebraic space over S .
- (2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ v finite type.

RNNが産み出した
数学論文モドキ

Stack Theory の教科書を
「学習」させたもの

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram



is a limit. Then \mathcal{G} is a finite type and assume S is a flat and \mathcal{F} and \mathcal{G} is a finite type f_* . This is of finite type diagrams, and

- the composition of \mathcal{G} is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

□

Proof. We have see that $X = \text{Spec}(R)$ and \mathcal{F} is a finite type representable by algebraic space. The property \mathcal{F} is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U .

Proof. This is clear that \mathcal{G} is a finite presentation, see Lemmas ??.
A reduced above we conclude that U is an open covering of \mathcal{C} . The fu
"field

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\mathbb{F}}^{-1}(\mathcal{O}_{X_{\text{étale}}}) \longrightarrow \mathcal{O}_{X'_t}^{-1} \mathcal{O}_{X_\lambda}(\mathcal{O}_{X'_t}^{\vee})$$

is an isomorphism of covering of \mathcal{O}_{X_t} . If \mathcal{F} is the unique element of \mathcal{F} such that X is an isomorphism.

The property \mathcal{F} is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme \mathcal{O}_X -algebra with \mathcal{F} are opens of finite type of \mathcal{C} . If \mathcal{F} is a scheme theoretic image points.

If \mathcal{F} is a finite direct sum \mathcal{O}_{X_λ} is a closed immersion, see Lemma ??
sequence of \mathcal{F} is a similar morphism.

RNNが産み出した
数学論文モドキ

Stack Theory の教科書を
「学習」させたもの

RNNは、Latexの構文規則を学習し、
こうしたシーケンスを構成できる。

```
\begin{proof}
We may assume that  $\mathcal{I}$  is an abelian sheaf on  $\mathcal{C}$ .
\item Given a morphism  $\Delta : \mathcal{F} \rightarrow \mathcal{I}$ 
is an injective and let  $\mathfrak{q}$  be an abelian sheaf on  $X$ .
Let  $\mathcal{F}$  be a fibered complex. Let  $\mathcal{F}$  be a category.
\begin{enumerate}
\item \hyperref[setain-construction-phantom]{Lemma}
\label{lemma-characterize-quasi-finite}
Let  $\mathcal{F}$  be an abelian quasi-coherent sheaf on  $\mathcal{C}$ .
Let  $\mathcal{F}$  be a coherent  $\mathcal{O}_X$ -module. Then
 $\mathcal{F}$  is an abelian catenary over  $\mathcal{C}$ .
\item The following are equivalent
\begin{enumerate}
\item  $\mathcal{F}$  is an  $\mathcal{O}_X$ -module.
\end{enumerate}
\end{lemma}
```

```
/*
 * Increment the size file of the new incorrect UL_FILTER group information
 * of the size generatively.
 */
static int indicate_policy(void)
{
    int error;
    if (fd == MARN_EPT) {
        /*
         * The kernel blank will coeld it to userspace.
         */
        if (ss->segment < mem_total)
            unblock_graph_and_set_blocked();
        else
            ret = 1;
        goto bail;
    }
    segaddr = in_SB(in.addr);
    selector = seg / 16;
    setup_works = true;
    for (i = 0; i < blocks; i++) {
        seq = buf[i++];
        bpf = bd->bd.next + i * search;
        if (fd) {
            current = blocked;
        }
    }
}
```

RNNが産み出した
Cプログラム・モドキ

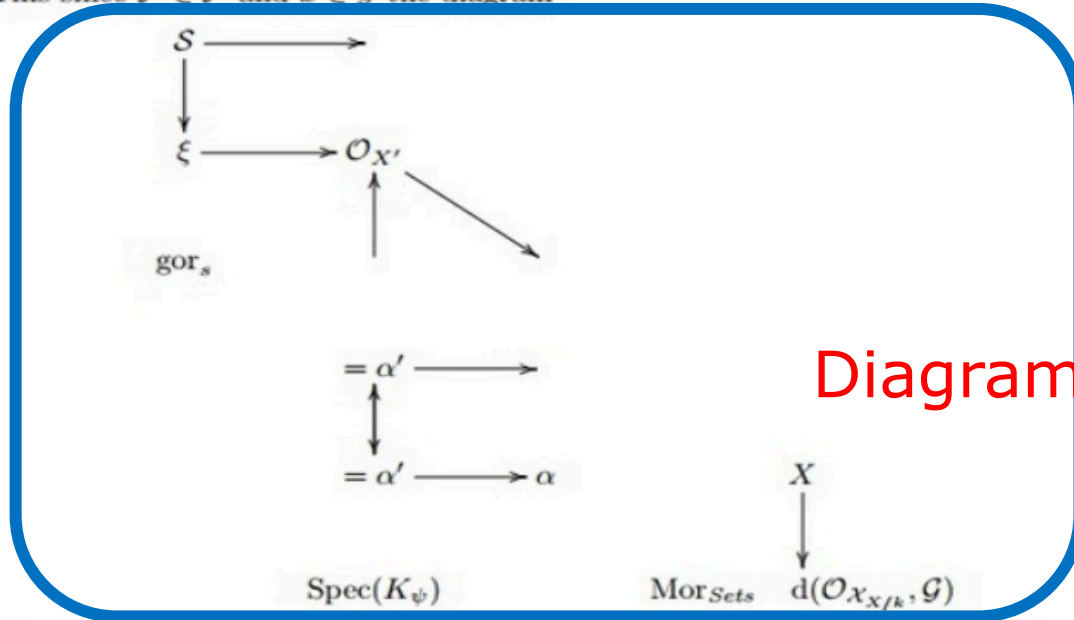
Linuxのソースコードを
「学習」させたもの

```
/*  
 * If this error is set, we will need anything right after that BSD.  
 */  
static void action_new_function(struct s_stat_info *wb)  
{  
    unsigned long flags;  
    int lel_idx_bit = e->edd, *sys & ~((unsigned long) *FIRST_COMPAT);  
    buf[0] = 0xFFFFFFFF & (bit << 4);  
    min(inc, slist->bytes);  
    printk(KERN_WARNING "Memory allocated %02x/%02x, "  
           "original MLL instead\n"),  
           min(min(multi_run - s->len, max) * num_data_in),  
           frame_pos, sz + first_seg);  
    div_u64_w(val, inb_p);  
    spin_unlock(&disk->queue_lock);  
    mutex_unlock(&s->sock->mutex);  
    mutex_unlock(&func->mutex);  
    return disassemble(info->pending_bh);  
}  
  
static void num_serial_settings(struct tty_struct *tty)  
{  
    if (tty == tty)  
        disable_single_st_p(dev);  
    pci_disable_spool(port);  
    return 0;  
}
```

RNNが産み出した
Cプログラム・モドキ

Linuxのソースコードを
「学習」させたもの

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram



Diagramの真似ができない！

is a limit. Then \mathcal{G} is a finite type and assume S is a flat and \mathcal{F} and \mathcal{G} is a finite type f_* . This is of finite type diagrams, and

- the composition of \mathcal{G} is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

Proof. We have see that $X = \text{Spec}(R)$ and \mathcal{F} is a finite type representable by algebraic space. The property \mathcal{F} is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U .

Proof. This is clear that \mathcal{G} is a finite presentation, see Lemmas ??.

A reduced above we conclude that U is an open covering of \mathcal{C} . The functor \mathcal{F} is a “field

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\bar{x}} \rightarrow \mathcal{O}_{X_{\text{étale}}} \longrightarrow \mathcal{O}_{X_{\lambda}}^{-1} \mathcal{O}_{X_{\lambda}}(\mathcal{O}_{X_{\eta}}^{\bar{v}})$$

is an isomorphism of covering of $\mathcal{O}_{X_{\lambda}}$. If \mathcal{F} is the unique element of \mathcal{F} such that X is an isomorphism.

The property \mathcal{F} is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme \mathcal{O}_X -algebra with \mathcal{F} are opens of finite type over S .

If \mathcal{F} is a scheme theoretic image points. □

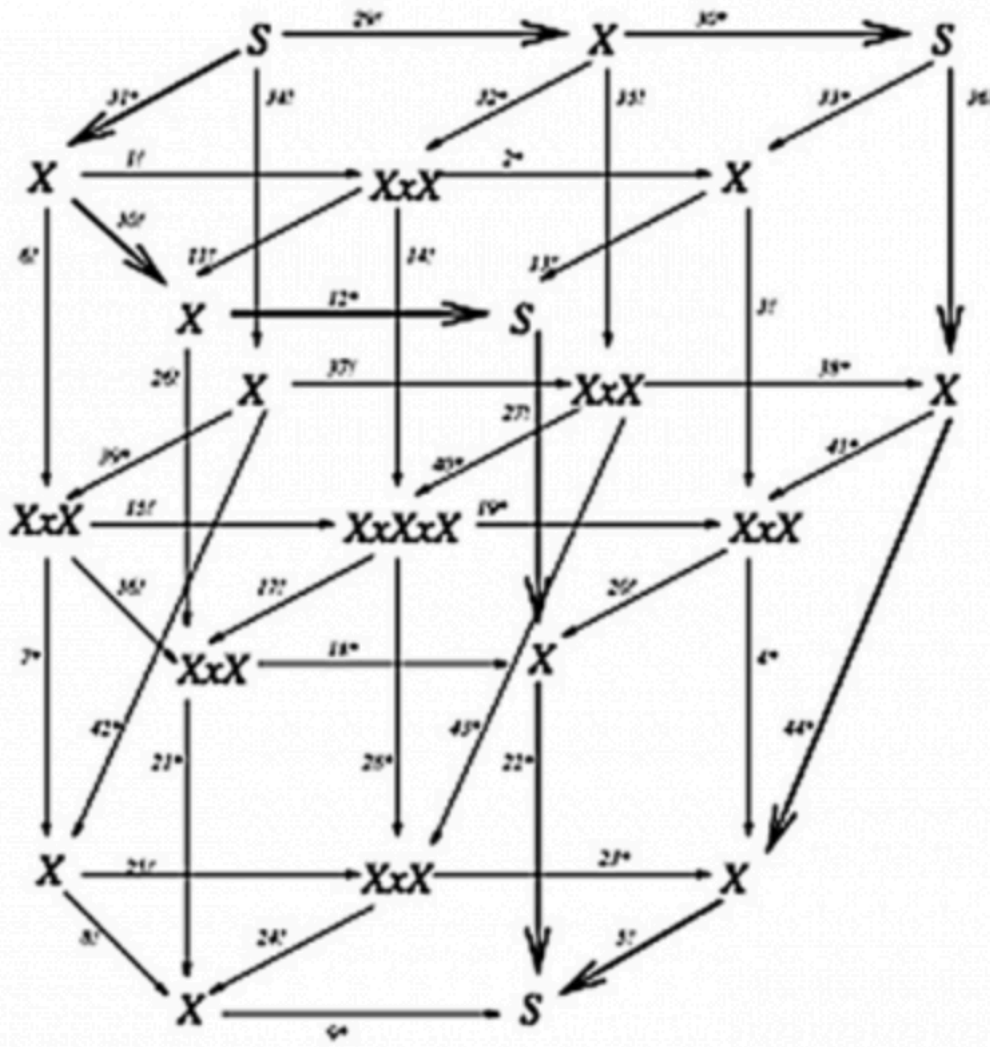
If \mathcal{F} is a finite direct sum $\mathcal{O}_{X_{\lambda}}$ is a closed immersion, see Lemma ??. This is a sequence of \mathcal{F} is a similar morphism.

RNNが産み出した
数学論文モドキ

Stack Theory の教科書を
「学習」させたもの

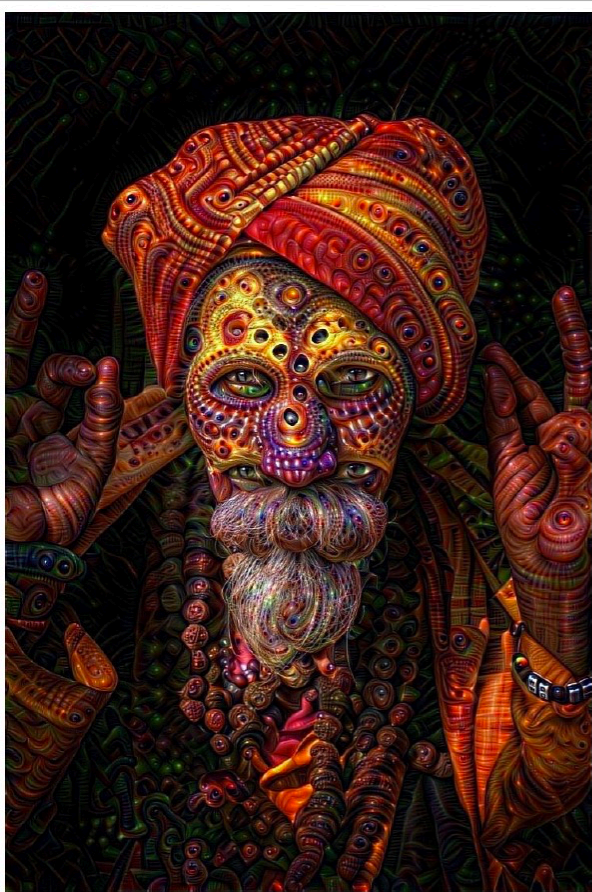
生成できない！

生成できる！



先の空白部分には、
こんな図形が入るはず

生成できる！



reddit.com/r/deepdream



LSTMの能力について

"Long Short Term Memory"

Sepp Hochreiter et al.

<https://goo.gl/sDAq81>

1997年

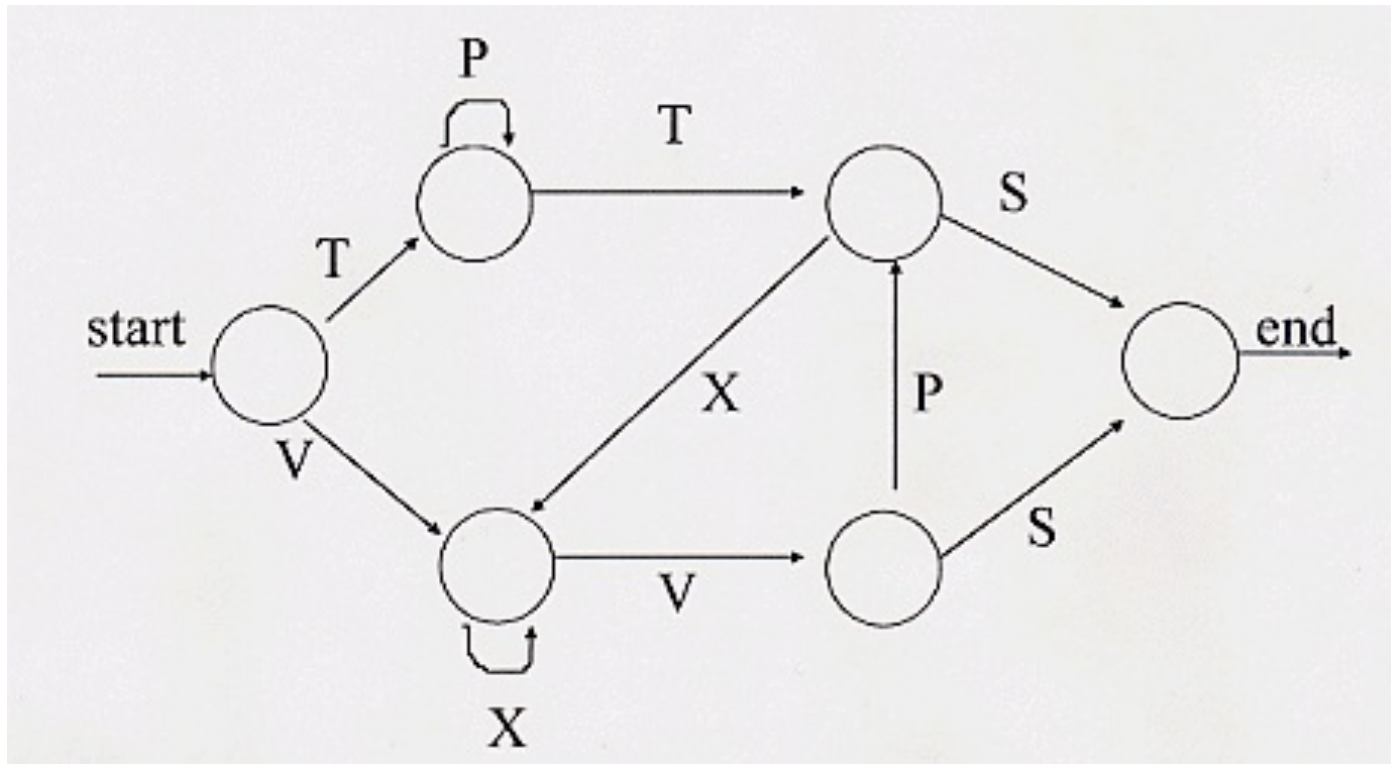


Hochreiterの90年代の発見

- LSTMの提案者であるHochreiterは、1990年代に、既に RNNが、極めて高い、認識能力を持つことを発見していた。LSTMの原論文に、詳細に、多数の実験結果が報告されている。ここでは、その中から、三つほどの例を紹介する。
- 一つは、有限オートマトンの生成する文字列の規則性を、LSTMが認識できるという実験結果である。これは、認知心理学の分野で、人間が持つ「人工的文法の学習 (Artificial grammar learning)」能力として研究されていたものである。 <https://goo.gl/VgvboK>
- もう一つは、LSTMが、掛け算を学習できるという実験である。ニューラルネットが、任意の関数の近似が可能であることは知られていたのだが、これはこれで興味ふかい実験である。
- 最後は、あるルールに従って、時間的に離れたところで起きるイベントのパターンを、LSTMが学習できるという実験である。これは、人間でも手こずりそうかもしれない。
- いずれの実験も、膨大な学習の繰り返しが必要である。

人工的な文法の学習

実験 1: REBER GRAMMAR



例えば、次の文字列、**VXVS**, **TPTXVS** は、文法にあってはいるが、**VXXS**, **TPTPS** は文法的ではない。

実験 1: Embedded Reber grammar

LSTMは、この文法を学習できる！

method	hidden units	# weights	learning rate	% of success	success after
RTRL	3	≈ 170	0.05	“some fraction”	173,000
RTRL	12	≈ 494	0.1	“some fraction”	25,000
ELM	15	≈ 435		0	>200,000
RCC	7-9	$\approx 119-198$		50	182,000
LSTM	4 blocks, size 1	264	0.1	100	39,740
LSTM	3 blocks, size 2	276	0.1	100	21,730
LSTM	3 blocks, size 2	276	0.2	97	14,060
LSTM	4 blocks, size 1	264	0.5	97	9,500
LSTM	3 blocks, size 2	276	0.5	100	8,440

掛け算の学習

実験5: MULTIPLICATION PROBLEM

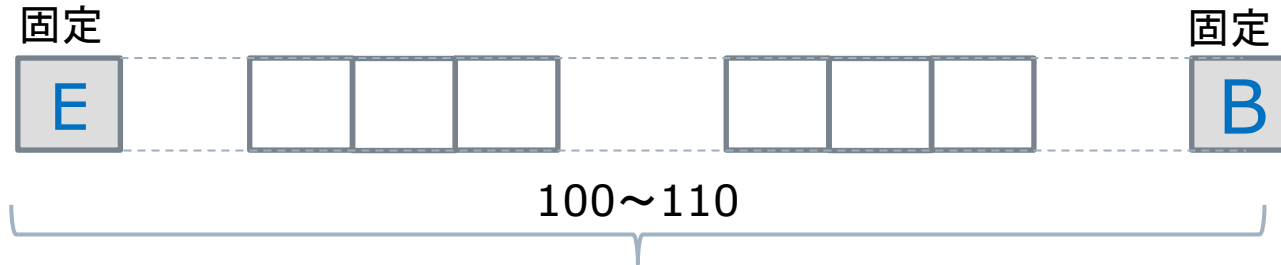
- **Task.** Like the task in Section 5.4, except that the first component of each pair is a real value randomly chosen from the interval $[0; 1]$. In the rare case where the first pair of the input sequence gets marked, we set X_1 to 1.0. The target at sequence end is the product $X_1 \times X_2$.

T	minimal lag	# weights	n_{seq}	# wrong predictions	MSE	Success after
100	50	93	140	139 out of 2560	0.0223	482,000
100	50	93	13	14 out of 2560	0.0139	1,273,000

LSTMは、足し算(実験4)も掛け算も学習できる！

実験6: TEMPORAL ORDER

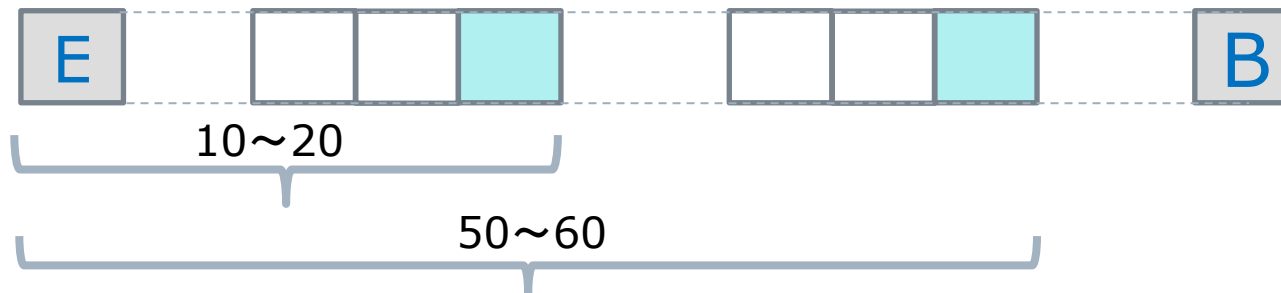
先頭にE、最後にBが入っている(固定)、長さ100~110(可変)の文字列がある。



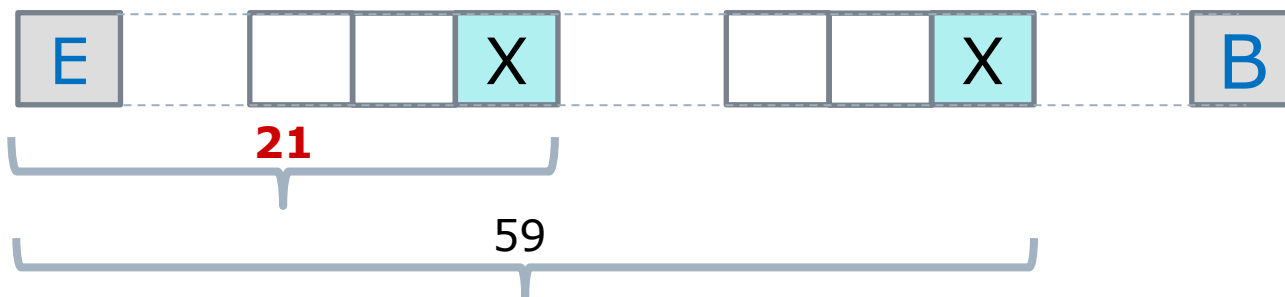
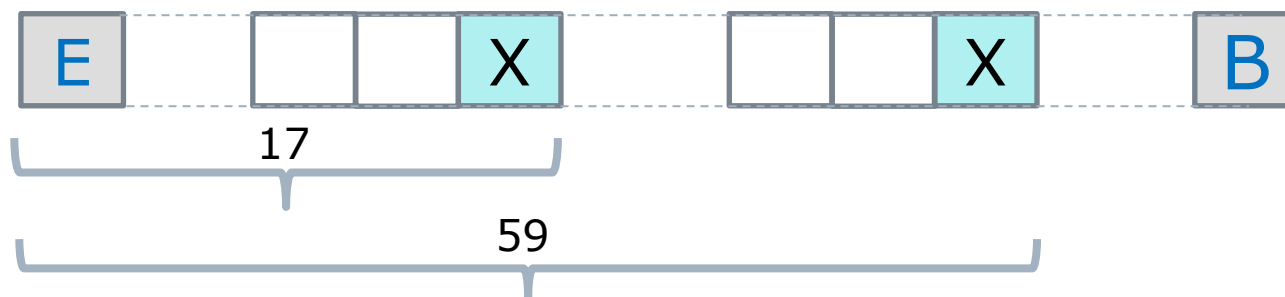
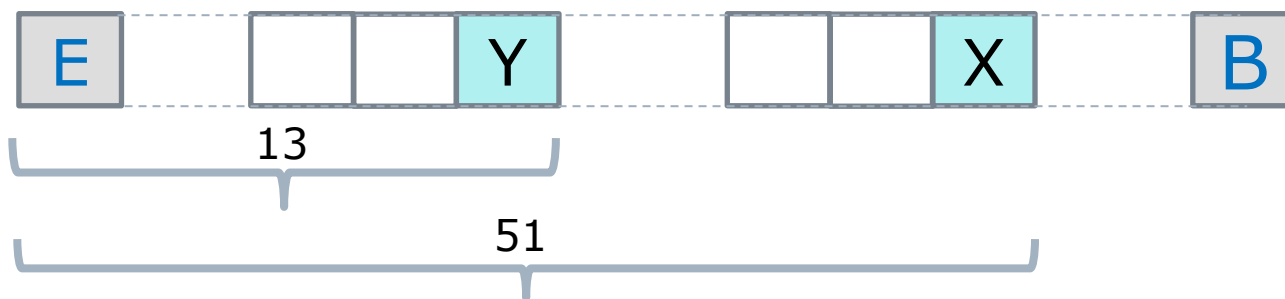
先頭のE、最後のB以外の場所には、a, b, c, dの文字がランダムに入っている。



ただし、先頭から10~20番目と50~60番目のランダムに選ばれた二箇所に、XまたはYの文字が入っている。



例えば、次の二例は、条件を満たすが、三番目の例は、条件を満たしていない。



この時、範囲指定された二箇所にも、
X,Xが入っているならQ、
X,Yが入っているならR、
Y,Xが入っているならS、
Y,Yが入っているならU
として、与えられた文字列を分類する。

先の第一例はS、第二例はQ、
第三例は、そのいずれでもないことになる。

task	# weights	# wrong predictions	Success after
Task 6a	156	1 out of 2560	31,390
Task 6b	308	2 out of 2560	571,100

Task6bは、三箇所版。

X,X,X -> Q; X,X,Y -> R; X,Y, X -> S; X,Y,Y -> U;

Y,X,X -> V ; Y,X,Y -> A; Y,Y,X -> B; Y,Y,Y -> C

の8分類。

2012年の爆発



Geoffrey E. Hinton



Yann LeCun



Yoshua Bengio

2012年に起きたこと

現在の人工知能に対する関心の高まりは、2012年に始まる。
この年、

1. Googleの16,000コアの巨大なシステム DistBeliefが、教師なしで**ネコ**と人間の顔を学習するのに成功した。
2. また、ImageNetの**画像認識**コンテストで、Convolutional Neural Network (CNN)のAlexNetが、圧勝した。
3. さらに、Tront大+Microsoft+Google+IBM 四者の共同研究チームが、**音声認識**にディープラーニングを用いた手法を公開。従来の方式を、はるかに上回る精度を達成した。

こうして、ニューラル・ネットワーク上のディープラーニングという手法は、現代のAI技術の中核の座を占めるに至った。

“Deep Neural Networks for Acoustic Modeling in Speech Recognition” Geoffrey Hinton et al.

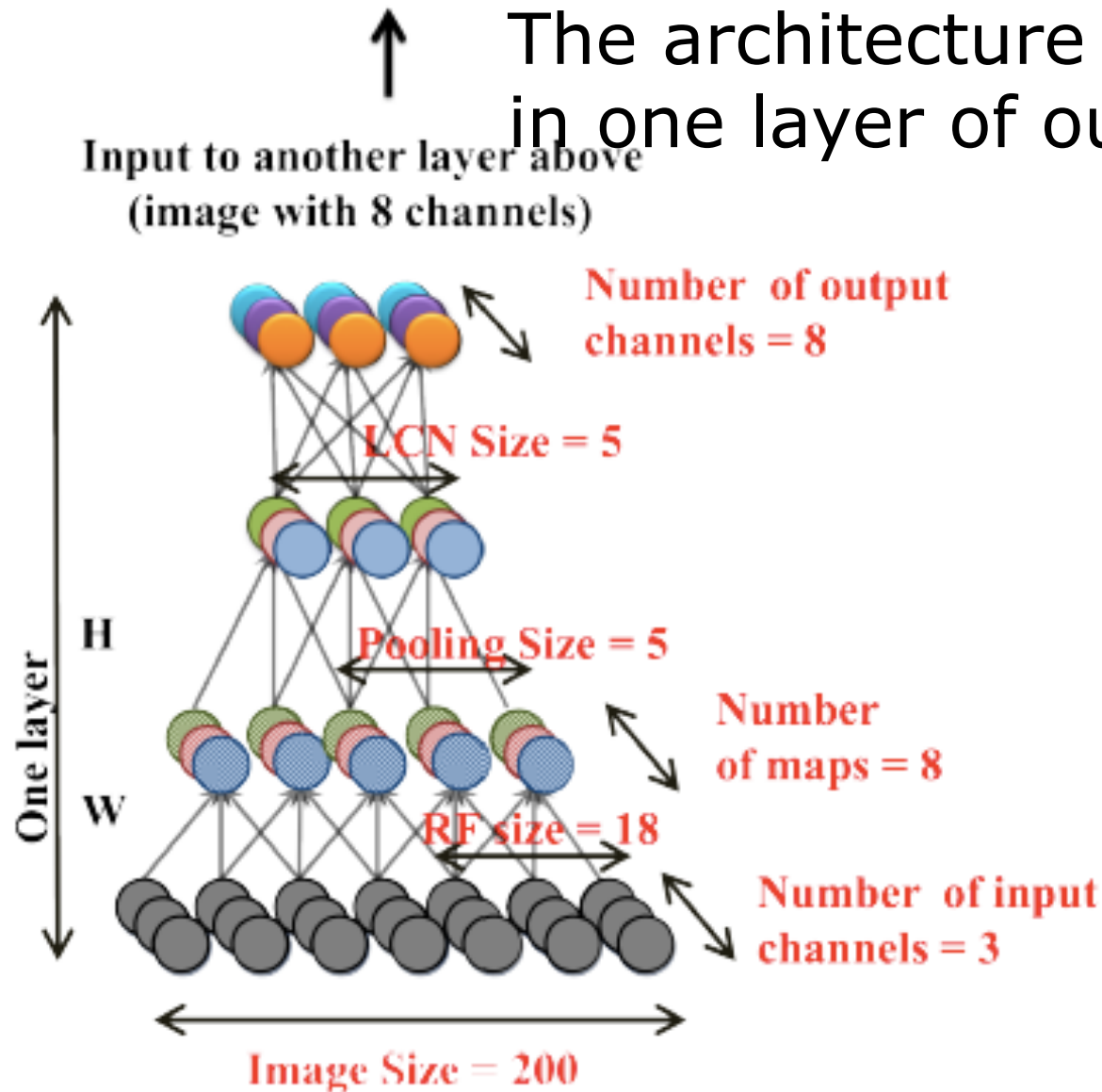
ディープラーニングを用いた
音声認識技術の飛躍

<https://goo.gl/GT6Bg9>

2012年



The architecture and parameters in one layer of our network.



Jeff Dean

- 我々は、ラベル付けされていないデータだけから、高レベルの、クラスに固有の特徴を検出することが出来るかという問題を考察した。例えば、ラベル付けされていないイメージだけから、顔の検出が可能かという問題である。
- この問題に答える為に、我々は、9層からなるローカルに疎に結合した、プーリングとローカルなコントラストの正規化の機能を備えた自動エンコーダを、大規模な画像データセット(モデルは10億のコネクションを持ち、データセットは、インターネットからダウンロードした200x200ピクセルの一千萬個のイメージからなる)上で訓練した。

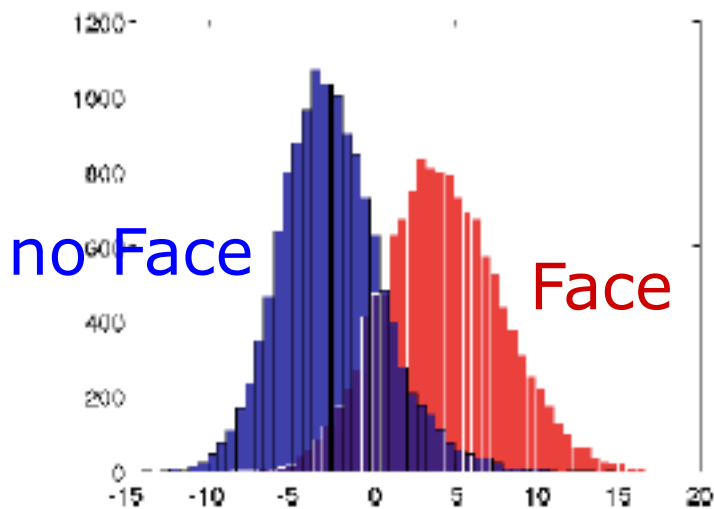
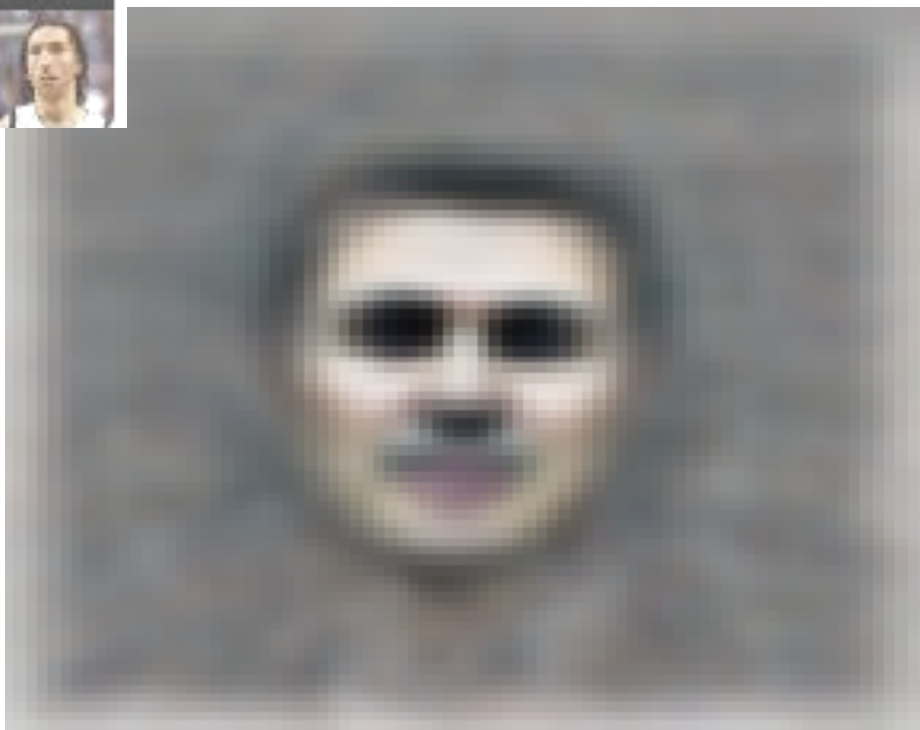
- 我々は、このネットワークを、1000台のマシン(16,000コア)のクラスター上で、並行計算モデルと非同期SGDを使って、三日間訓練した。
- 広く受け入れられているように見える直観に反して、我々の実験結果は、顔であるかそうでないかのラベルをイメージにつける必要なしに、顔の検出が可能であることを明らかにした。
- コントロールの実験では、この特徴の検出は、変換だけでなく、拡大や画面外への回転に対しても、頑健であることを示した。

Googleの猫

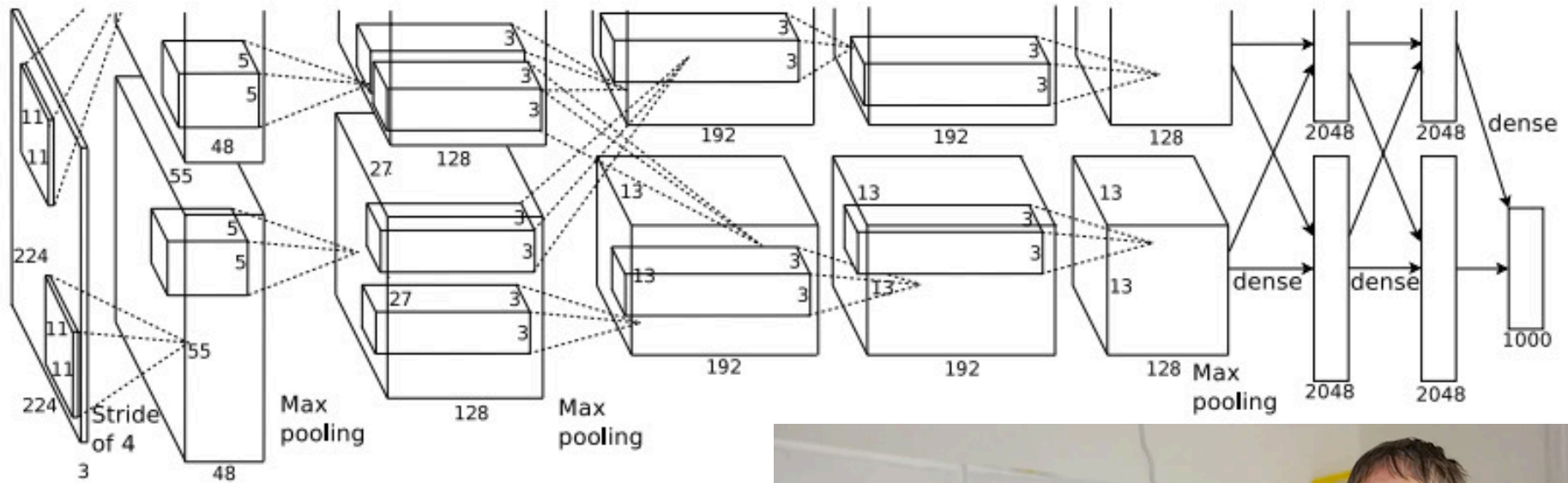


2012年6月

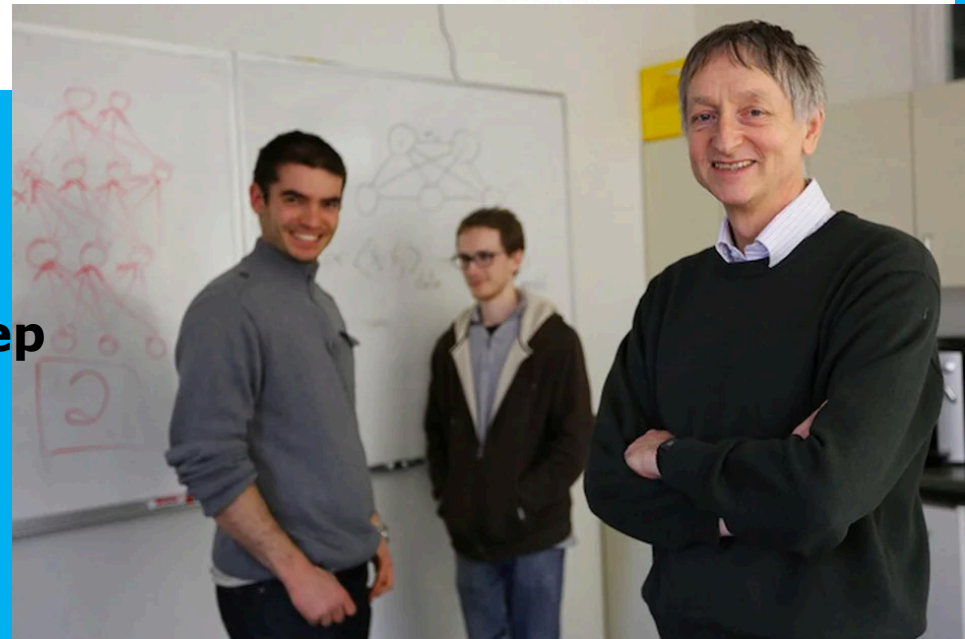
これが、一千万個の画像データから、16,000コアのマシンが三日間かけて抽出した「人間の顔」のイメージらしい。子供の絵に似ているのが面白い。



AlexNet



ImageNet Classification with Deep Convolutional Neural Networks



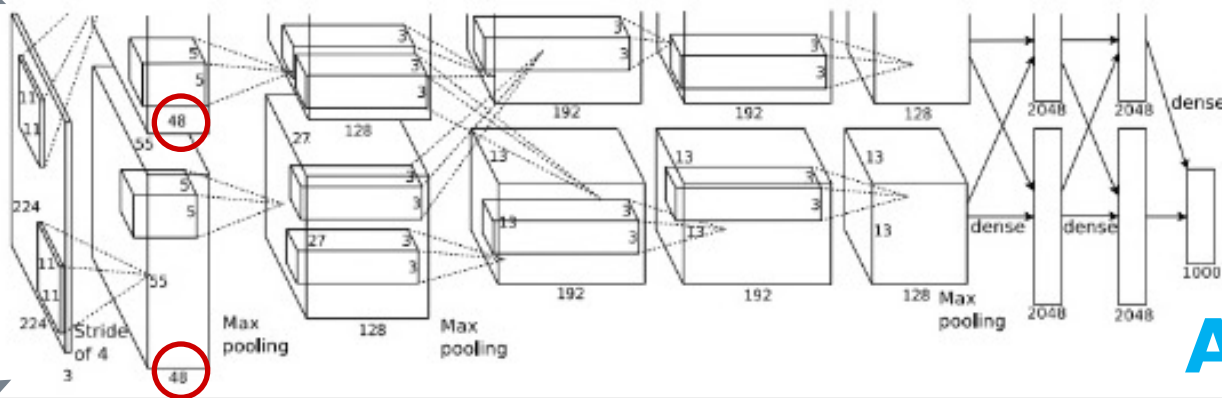
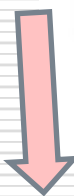
CNNで学習されたフィルター



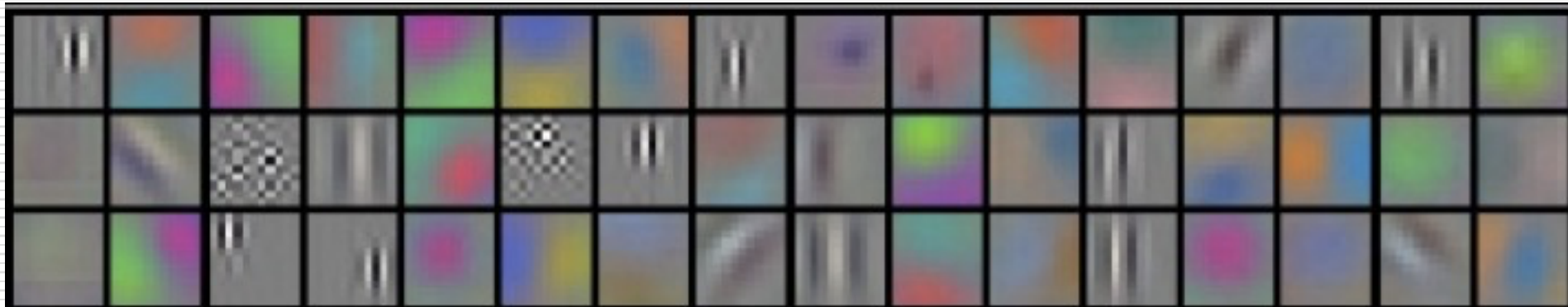
上半分で学習された48個のフィルター



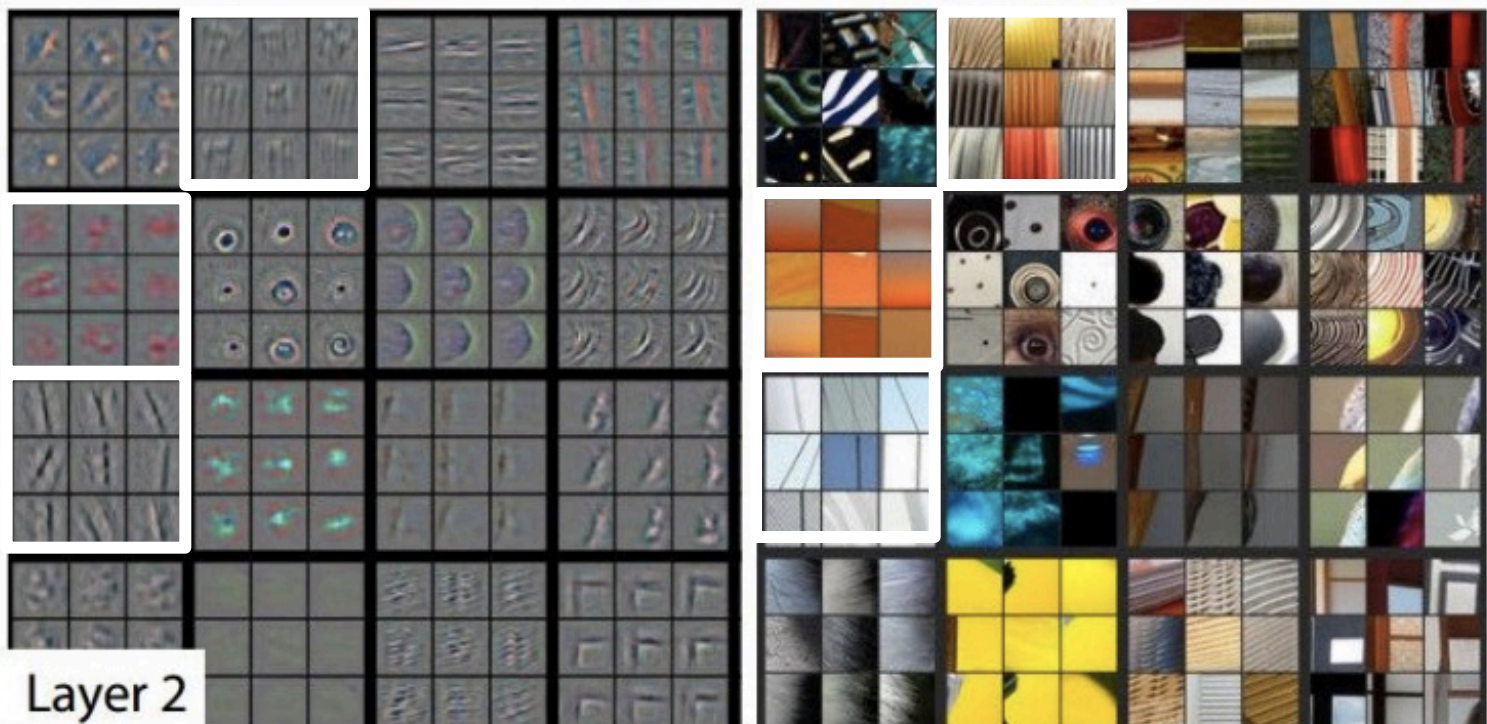
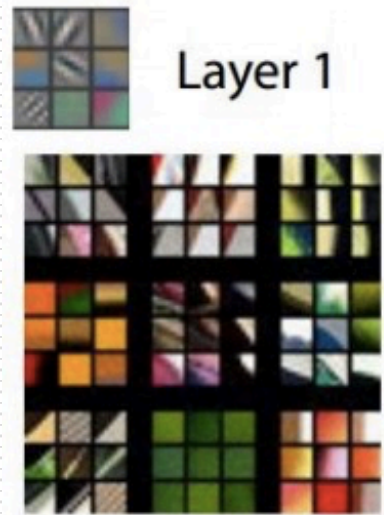
下半分で学習された48個のフィルター



AlexNet

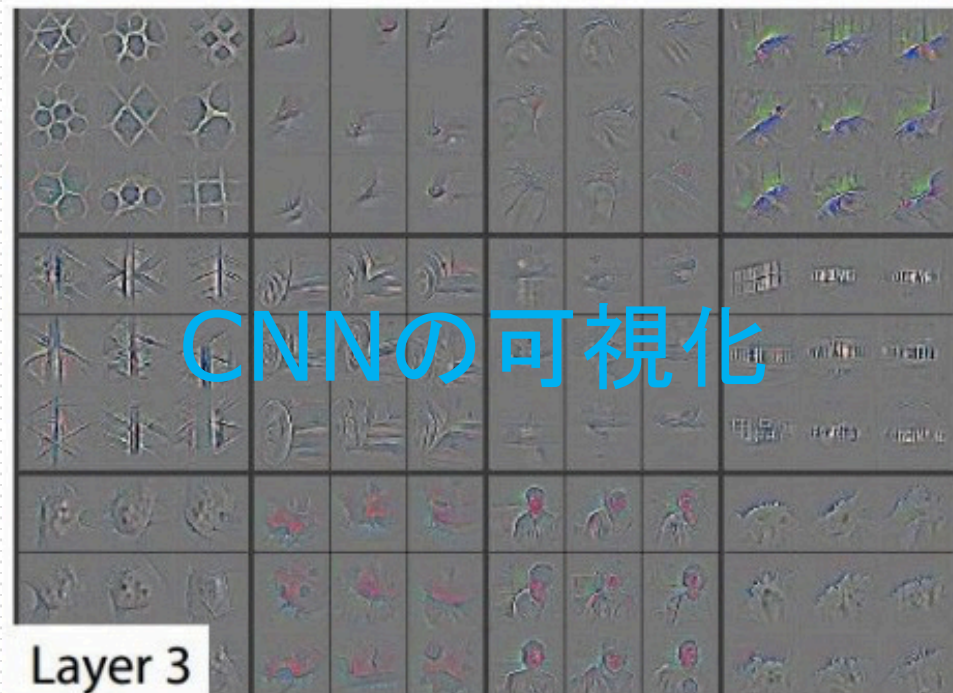


Layer 1



Layer 2

フィルターと
対応する画像

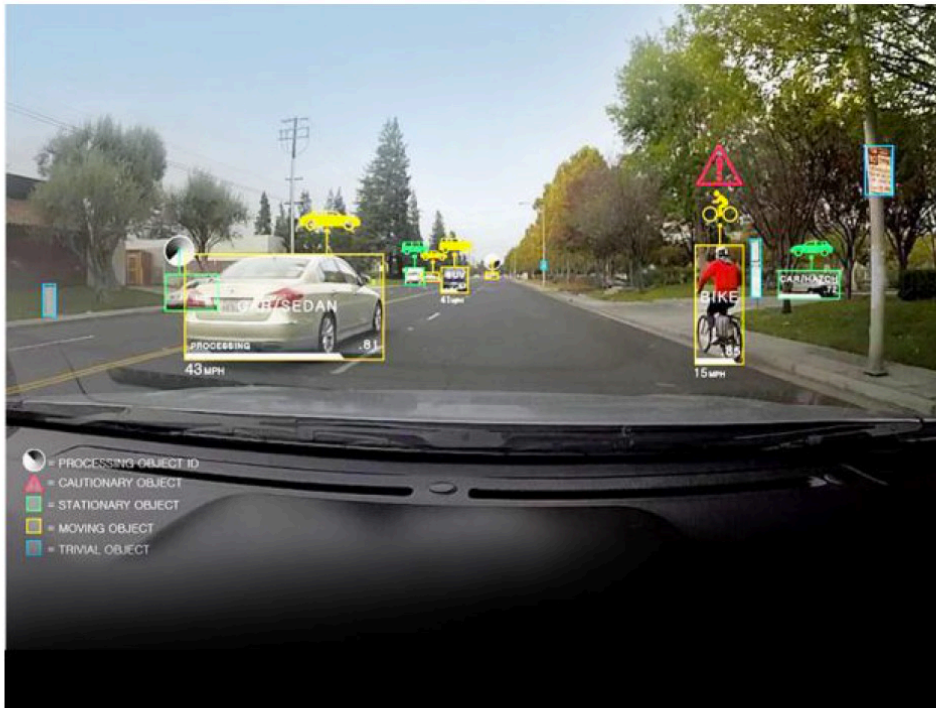


Layer 3

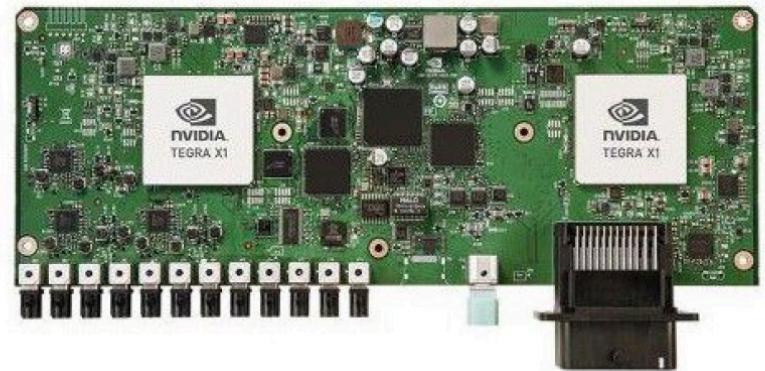


CNNの可視化

Fast-forward to today: ConvNets are everywhere



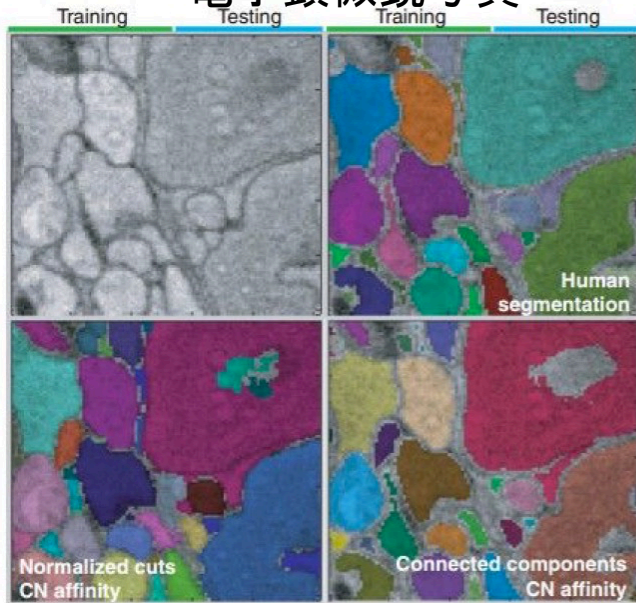
self-driving cars



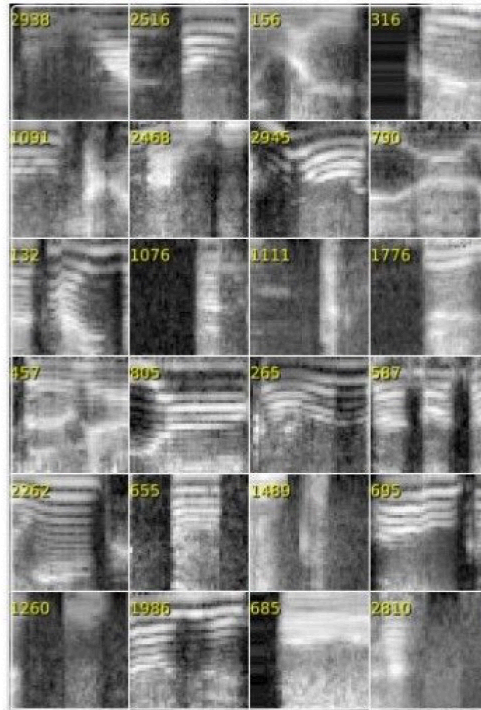
NVIDIA Tegra X1

Fast-forward to today: ConvNets are everywhere

ウサギの網膜の 電子顕微鏡写真



[Turaga et al., 2010]



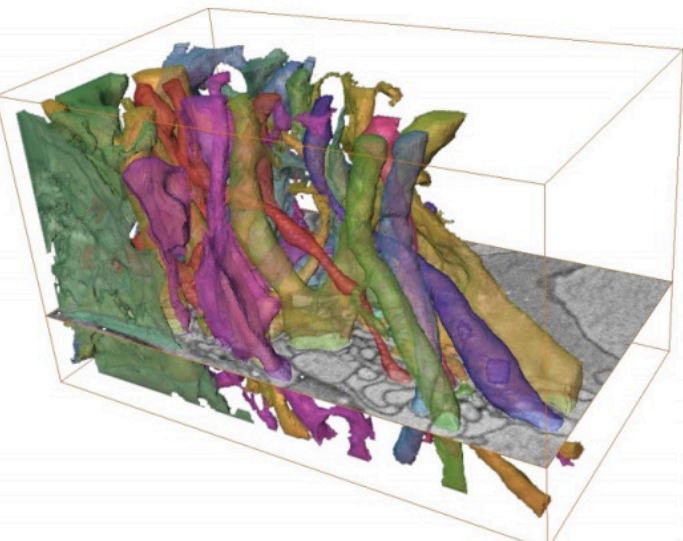
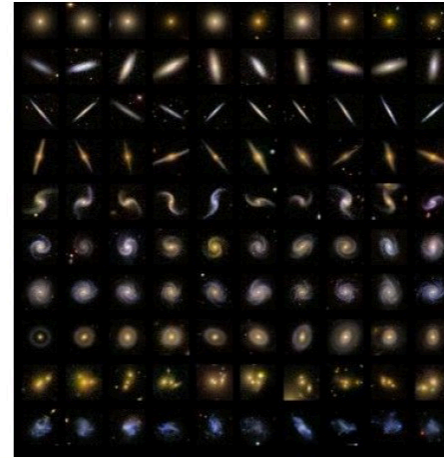
I caught this movie on the Sci-Fi channel recently. It actually turned out to be pretty decent as far as B-list horror/suspense films go. Two guys (one naive and one used) resulted in a road trip to stop a wedding but have the worst possible luck when a maniac in a freaky make-shift tank/truck hybrid decides to play cat-and-mouse with them. Things are further complicated when they pick up a ridiculously whoosh hitchhiker. What makes this film unique is that the combination of comedy and terror actually work in this movie, unlike so many others. The two guys are likable enough and there are some good chase/suspense scenes. Nice pacing and comic timing make this movie more than passable for the horror/slasher buff. **Definitely worth checking out!**

I just saw this on a local independent station in the New York City area. The cast showed promise but when I saw the director, George Cosmatos, I became suspicious. And sure enough, it was every bit as bad, every bit as pointless and stupid as every George Cosmatos movie I ever saw. He's like a stupid man's Michael Bay - with all the awfulness that accolade promises. There's no point to the conspiracy, no burning issues that urge the conspirators on. We are left to ourselves to connect the dots from one bit of graffiti on various walls in the film to the next. Thus, the current budget crisis, the war in Iraq, Islamic extremism, the fate of social security, 47 million Americans without health care, stagnating wages, and the death of the middle class are all subsumed by the sheer terror of graffiti. A truly, stunningly idiotic film.

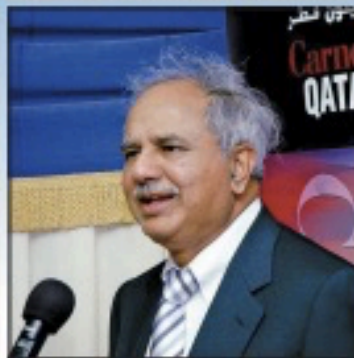
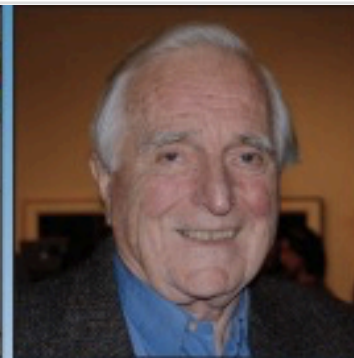
Graphics is far from the best part of the game. **This is the number one best TH game in the series.** Next to Underground. It deserves strong love. It is an insane game. There are massive levels, massive unlock-able characters... It's just a massive game. **Waste your money on this game.** This is the kind of money that is wasted properly. And even though graphics suck, that doesn't make a game good. Actually, the graphics were good at the time. Today the graphics are crap. WHO CARES? As they say in Canada, This is the fun game, aye. (You get to go to Canada in THPS3) Well, I don't know if they say that, but they might, who knows. Well, Canadian people do. Wait a minute, I'm getting off topic. This game rocks. Buy it, play it, enjoy it, love it. It's PURE BRILLIANCE.

The first was good and original. I was a net bad horror/comedy movie. So I heard a second one was made and I had to watch it. What really makes this movie work is Judd Nelson's character and the sometimes clever script. **A pretty good script for a person who wrote the Final Destination films and the direction was okay.** Sometimes there's scenes where it looks like it was filmed using a home video camera with a grainy look. Great made-for-TV movie. **It was worth the rental and probably worth buying just to get that nice eerie feeling and watch Judd Nelson's Stanley doing what he does best.** I suggest newcomers to watch the first one before watching the sequel, just so you'll have an idea what Stanley is like and get a little history background.

[Denil et al. 2014]



個人的に印象的だったこと



人工知能の殿堂 by IEEE 2010年

H a l l o f F a m e

AI's Hall of Fame

2010年 IEEE特集

AI's Hall of Fame

- **Edward Feigenbaum**
Engineering Knowledge
- **John McCarthy**
Logic and Common Sense
- **Marvin Minsky**
Cognitive Science and AI
- **Douglas Engelbart**
AI and Interactive Computing
- **Tim Berners-Lee**
Collective Knowledge
- **Lotfi Zadeh**
Fuzzy Logic and Computational Intelligence
- **Noam Chomsky**
Computational Linguistics and Cognitive Science
- **Raj Reddy**
AI and Societal Impact
- **Judea Pearl**
Probability, Causality, and Intelligence
- **Nils J. Nilsson**
Problem Solving and Planning

Team MIT: Atlas Robot Push Recovery

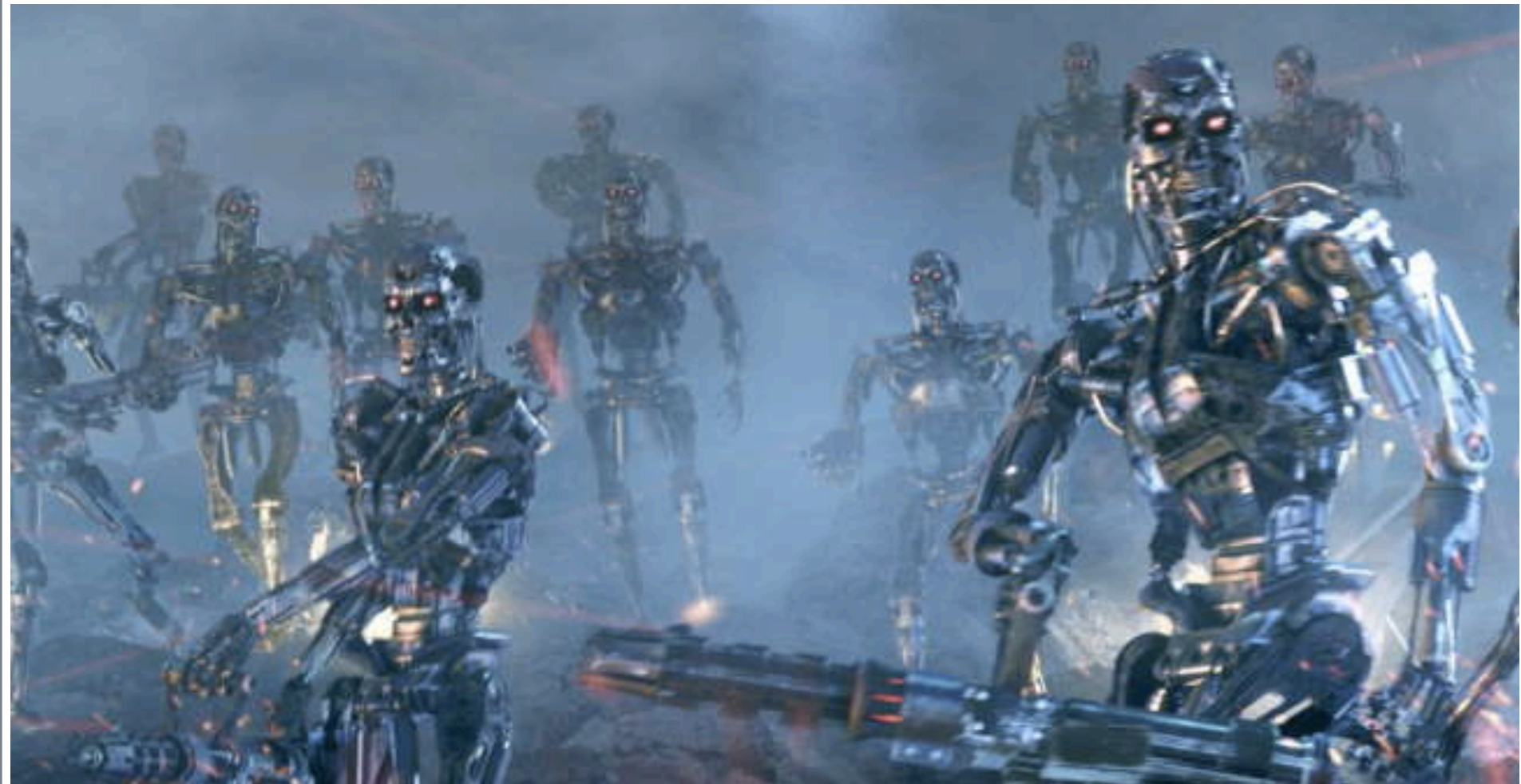


<https://www.youtube.com/watch?t=193&v=2mhIYetkNIg>

historyBuff.speechAsset. **SampleUtterances.txt**

```
GetFirstEventIntent {january twenty sixth|day}  
GetFirstEventIntent {january twenty seventh|day}  
GetFirstEventIntent {january twenty eighth|day}  
GetFirstEventIntent {january twenty ninth|day}  
GetFirstEventIntent {january thirtieth|day}  
GetFirstEventIntent {january thirty first|day}  
GetFirstEventIntent {february first|day}  
GetFirstEventIntent {february second|day}  
GetFirstEventIntent {february third|day}  
GetFirstEventIntent {february fourth|day}  
GetFirstEventIntent {february fifth|day}  
GetFirstEventIntent {february sixth|day}  
GetFirstEventIntent {february seventh|day}  
GetFirstEventIntent {february eighth|day}  
GetFirstEventIntent {february ninth|day}  
GetFirstEventIntent {february tenth|day}  
GetFirstEventIntent {february eleventh|day}  
GetFirstEventIntent {february twelfth|day}
```

“Elon Musk lives in fear of Google's killer robot army”





Predicted Tags:

portrait

eyeglasses

north america

promotion

one

facial expression

red carpet

men

adult

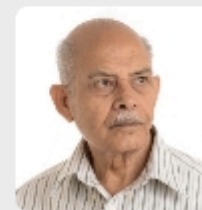
people

Stats:

Size: 76.15 KB

Time: 59 ms

Similar Images:





Predicted Tags:

portrait

eyeglasses

north america

promotion

one

facial expression

red carpet

men

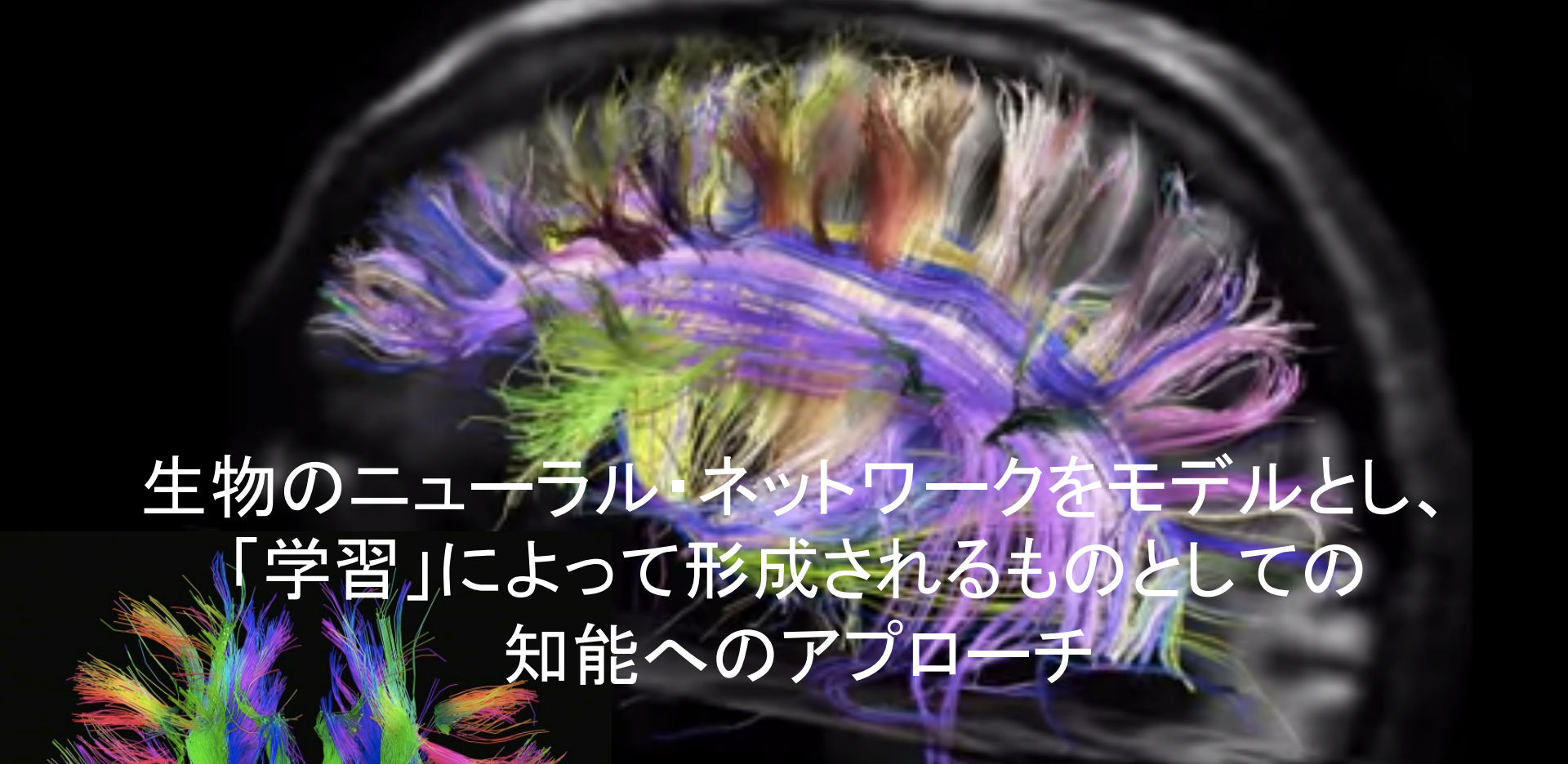
adult

?

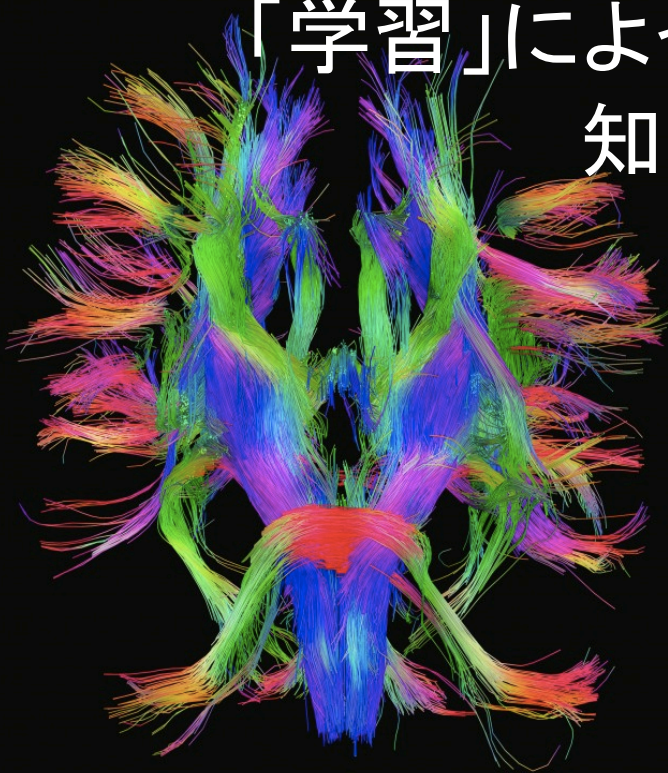


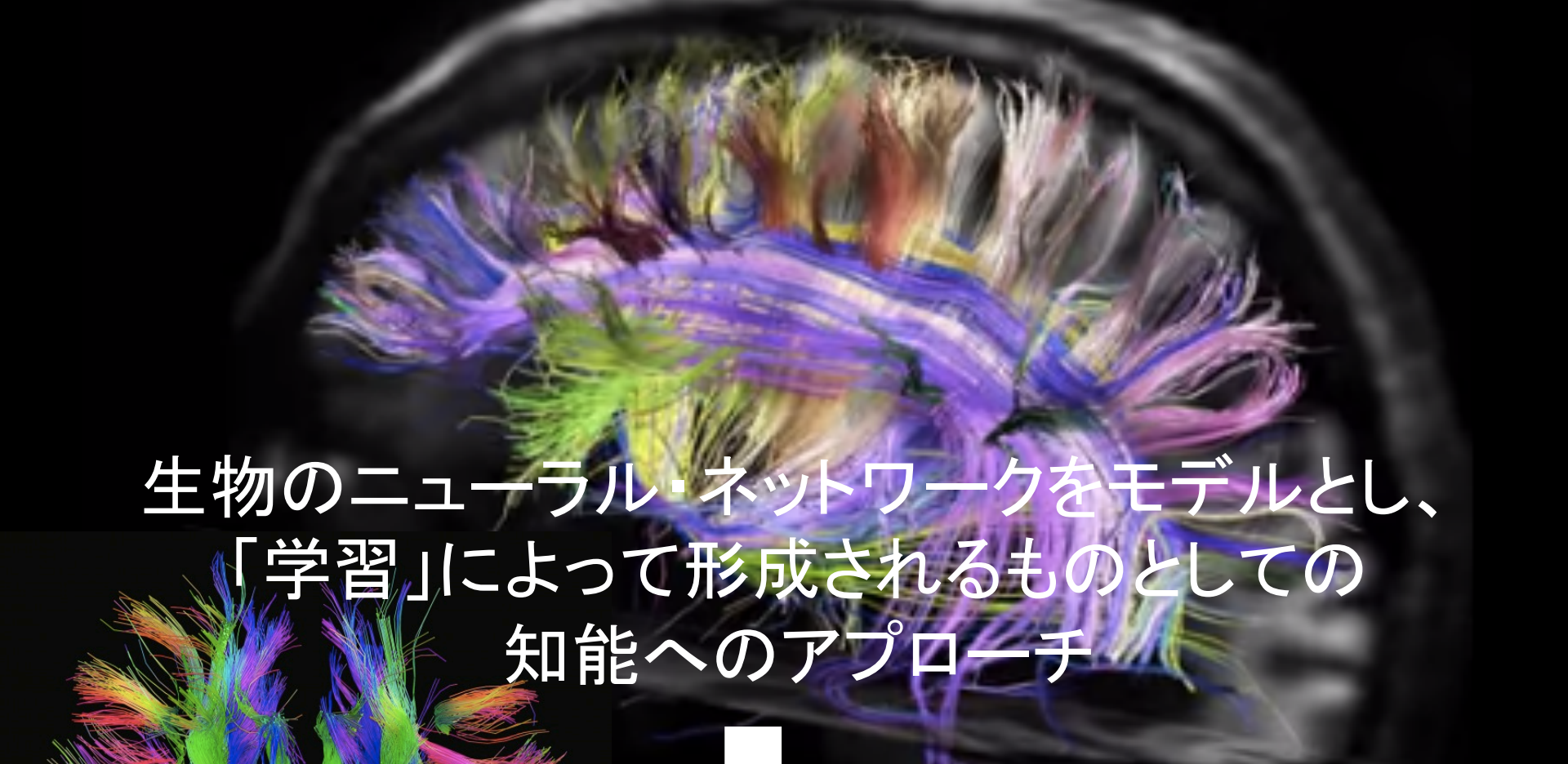
「丸山さんの顔の輪郭。右頬のラインはカリフォルニアと完全一致です。」

「機械の思考」についての 二つのアプローチ



生物のニューラル・ネットワークをモデルとし、
「学習」によって形成されるものとしての
知能へのアプローチ

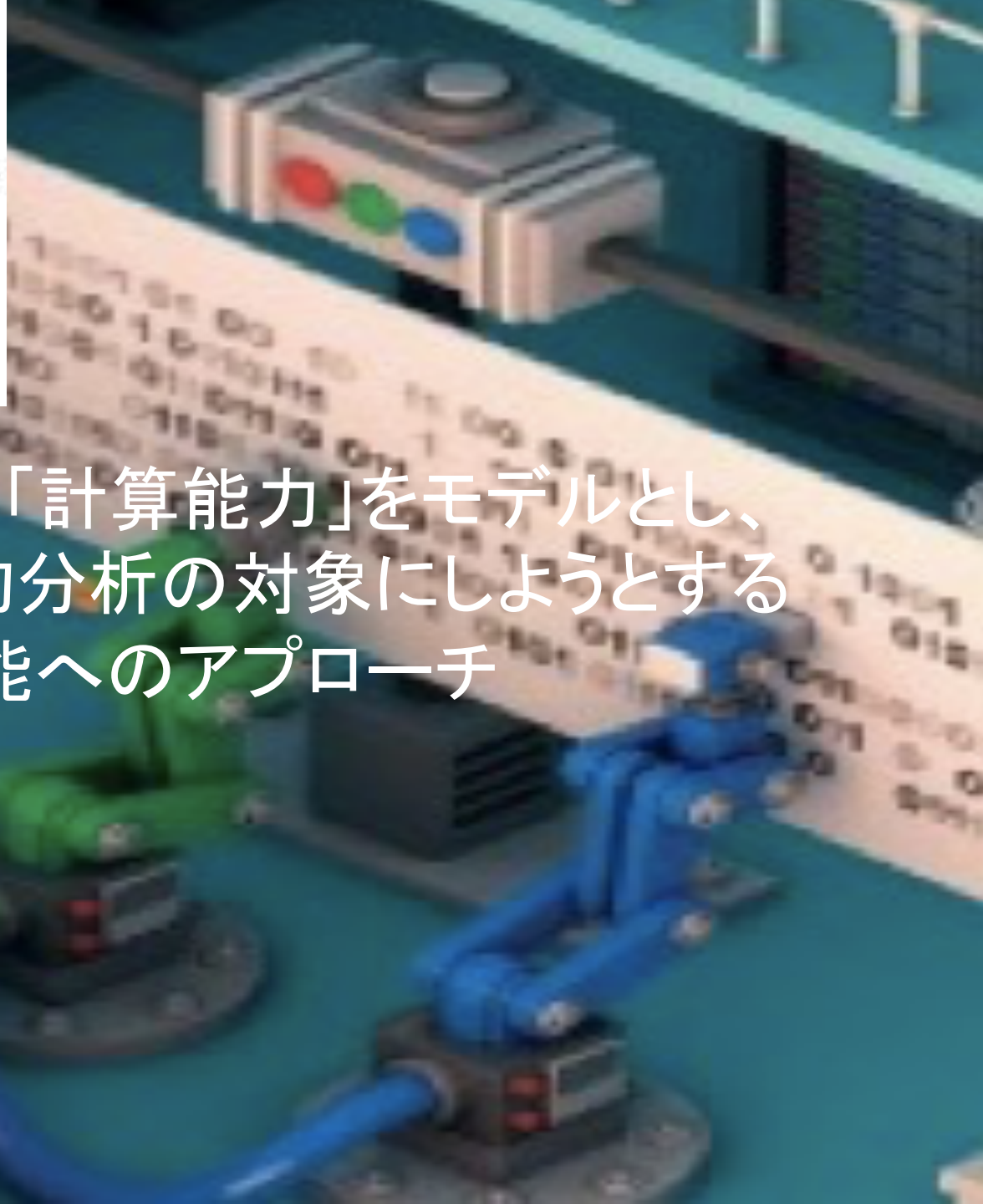
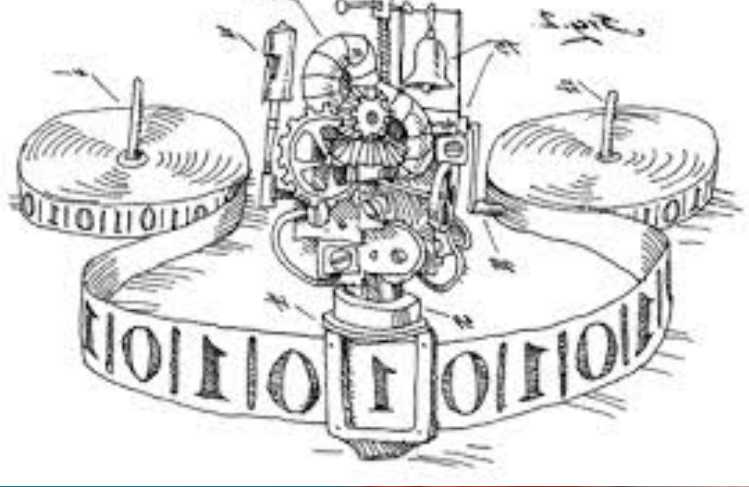




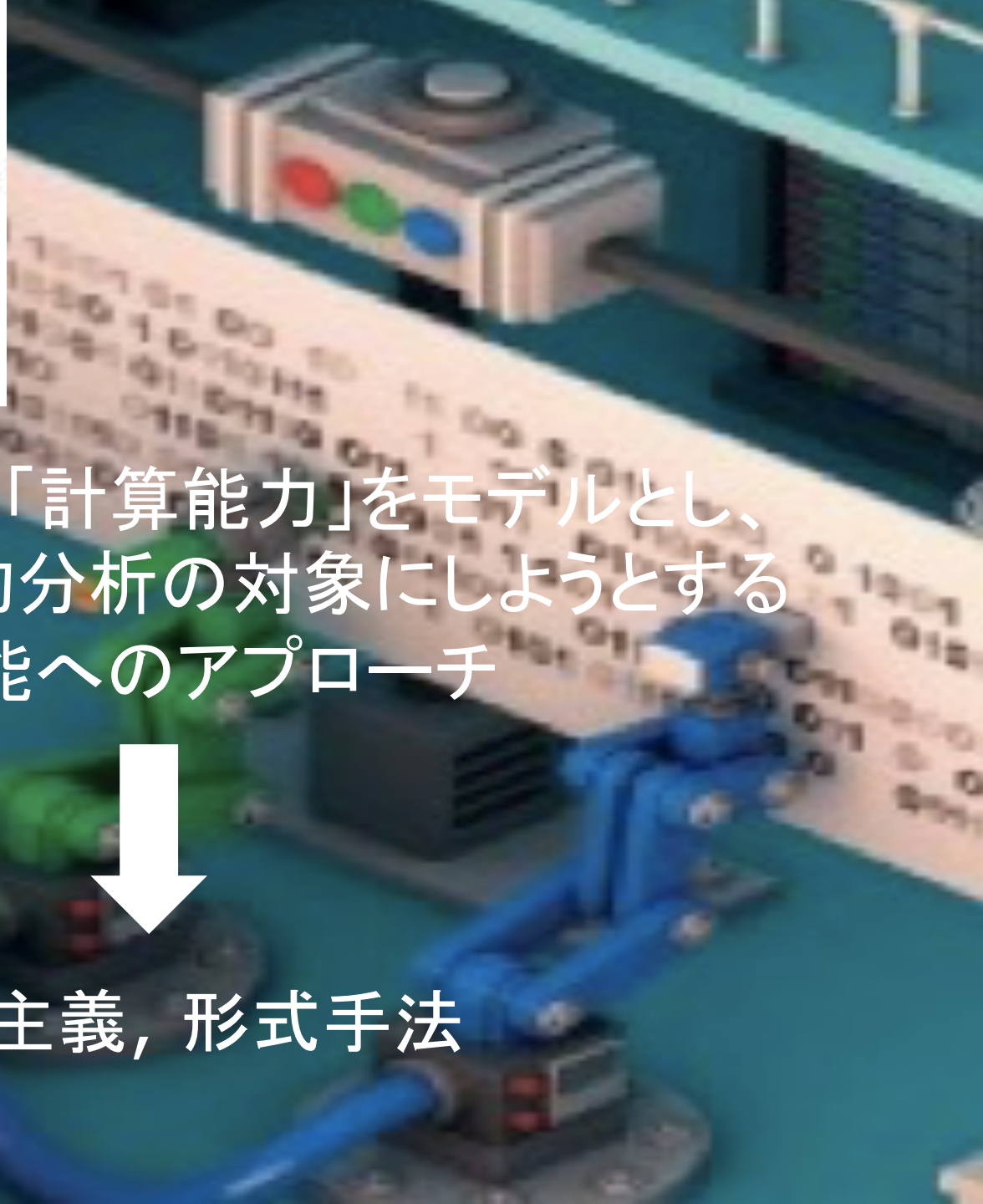
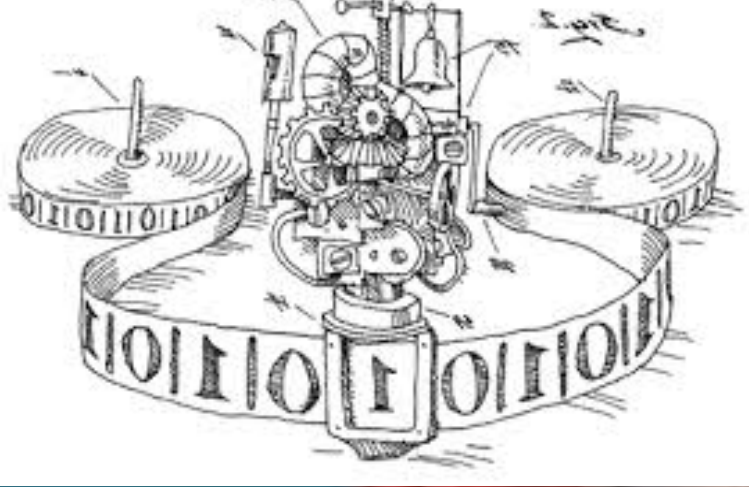
生物のニューラル・ネットワークをモデルとし、
「学習」によって形成されるものとしての
知能へのアプローチ



コネクショニズム、Deep Learning



人間・機械の「計算能力」をモデルとし、
知能を数学的分析の対象にしようとする
知能へのアプローチ



人間・機械の「計算能力」をモデルとし、
知能を数学的分析の対象にしようとする
知能へのアプローチ



計算主義, 形式手法

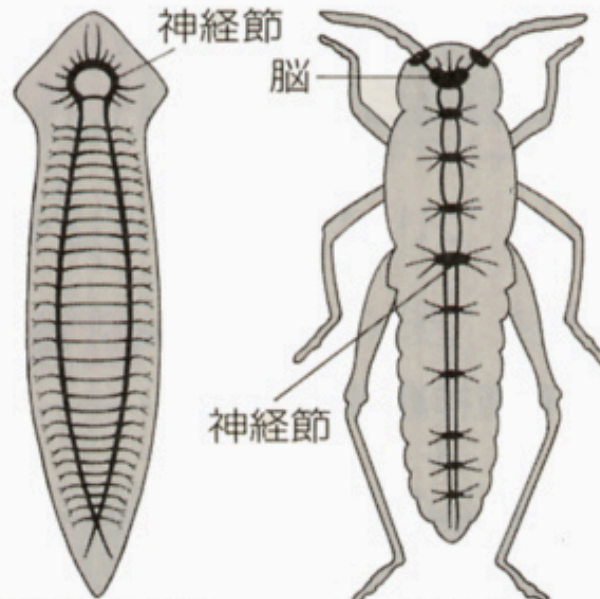


生物のニューラル・ネットワークを
モデルとするアプローチ

クラゲだって神経はある

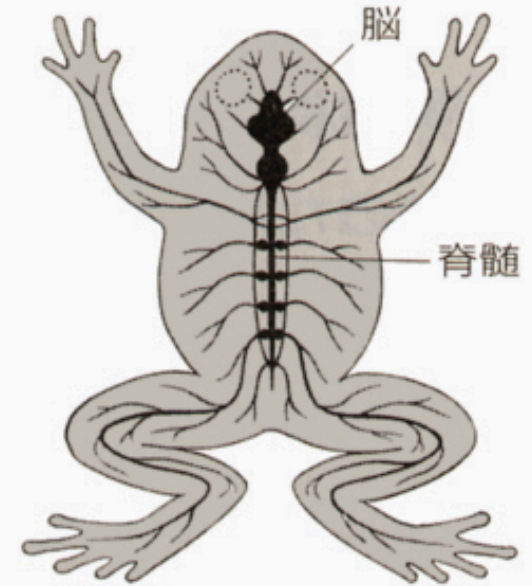


ヒドラ
(散在神経系)



プラナリア
(かご形神経系)

バッタ
(はしご形神経系)



カエル
(管状神経系)

<http://www.tmd.ac.jp/artsci/biol/textlife/neuron.htm>

「神経系の起源と進化」

<http://bit.ly/1qR2Dmq>

NEURONS AND SYNAPSES
 This wiring diagram shows more than 4,500 of the 8,000 neuron-to-neuron connections, or synapses, in the worm's nervous system. Each dot represents a single neuron:

- **SENSORY NEURONS**
detect external stimuli
- **CONNECTOR NEURONS**
relay signals
- **MOTOR NEURONS**
control muscle movement

線虫のC. Eleganceは、すべての神経の接続がわかっている唯一の生物である。
302の神経と8,000のシナプスがある。

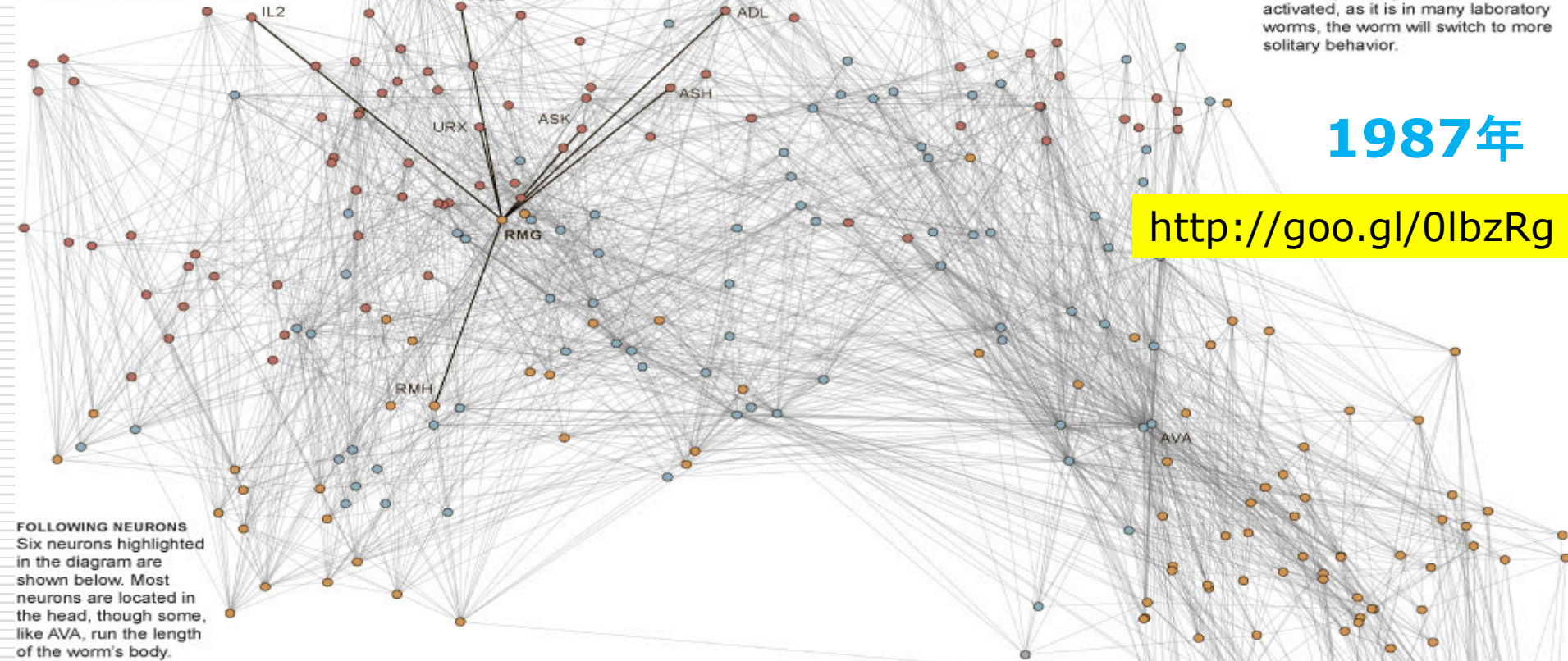
SOCIAL AND SOLITARY WORMS
 Most worms in nature congregate in clumps, a behavior controlled by two neurons called RMG, one of which is highlighted at left.

RMG combines information from several sensory neurons, also highlighted, and then signals the worm's muscles to move toward nearby worms if conditions are right.

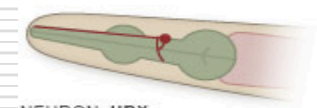
But if a specific gene in RMG is activated, as it is in many laboratory worms, the worm will switch to more solitary behavior.

1987年

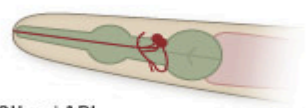
<http://goo.gl/0lbzRg>



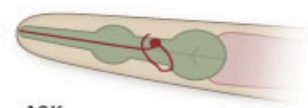
FOLLOWING NEURONS
 Six neurons highlighted in the diagram are shown below. Most neurons are located in the head, though some, like AVA, run the length of the worm's body.



NEURON: URX
 Senses oxygen in the environment



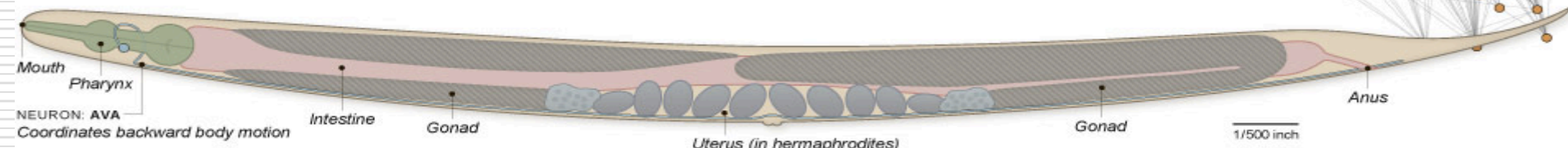
ASH and ADL
 Sense poisons and other noxious stimuli



ASK
 Senses food signals and detects sex pheromones from other worms

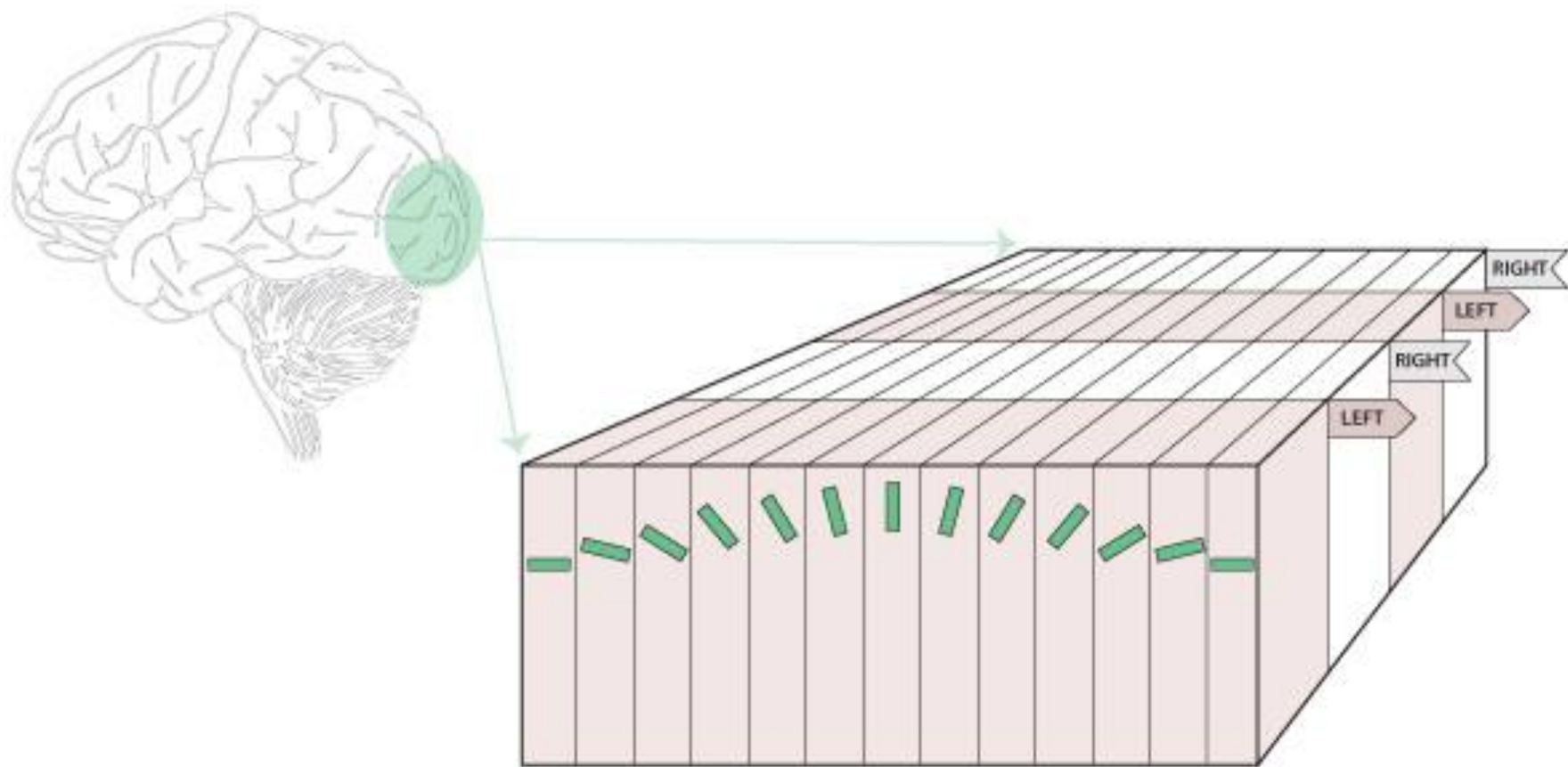


RMG
 Hub of a network of neurons that controls social behavior



NEURON: AVA
 Coordinates backward body motion

60年代から70年代にかけての、Hubel と Wieselの
大脳視覚野の研究は、各方面に大きな影響を与えた。
ニューラル・ネットワークの最初の研究も、こうした影響の
もとで始まった。

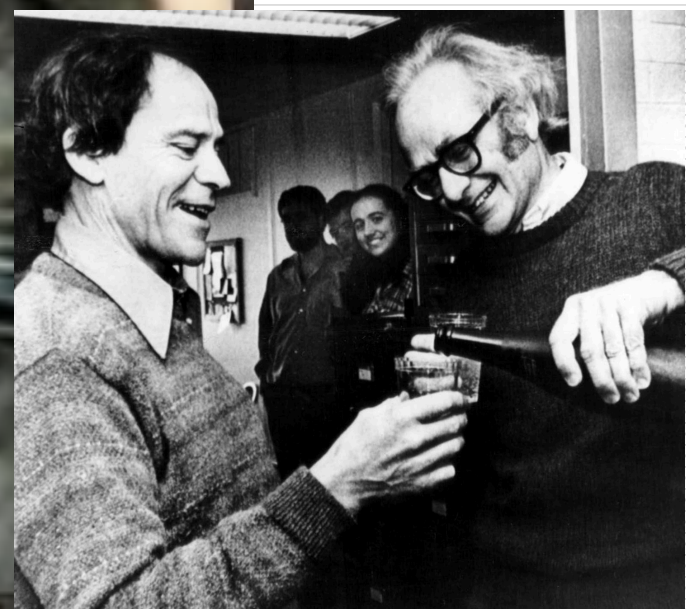




1960年代

David H. Hubel

Torsten Wiesel



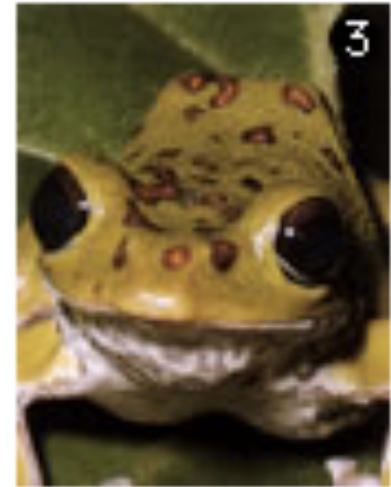
アノマロカリス



「視覚」の誕生と発展

- 生物の進化の歴史の中で、「視覚」の能力は、5億年前の「カンブリア紀」に獲得された。それは、長い長い歴史を持っている。当時の食物連鎖の頂点に立っていた「アノマロカリス」は、立派な眼を持っていた。
- 「眼」の獲得、同じことだが、「視覚」の能力の獲得は、「捕食者」にとっても（獲物を見つける）、「被捕食者」にとっても（捕食者から逃れる）、その「生き残り」に、死活的に重要な意味を持つことになった。人間を含めて、すべての動物で、強い淘汰の圧力のもとで、視覚処理の能力は、高度に発達している。
- 同時に、この「視覚」の能力は、追いかける、捉える、あるいは、逃げるといった「運動能力」に深く結びついている

多くの生物は目を持っている



ゲーリングによる「目の遺伝子」Pax-6の発見

https://www.brh.co.jp/seimeishi/journal/012/ss_1.html



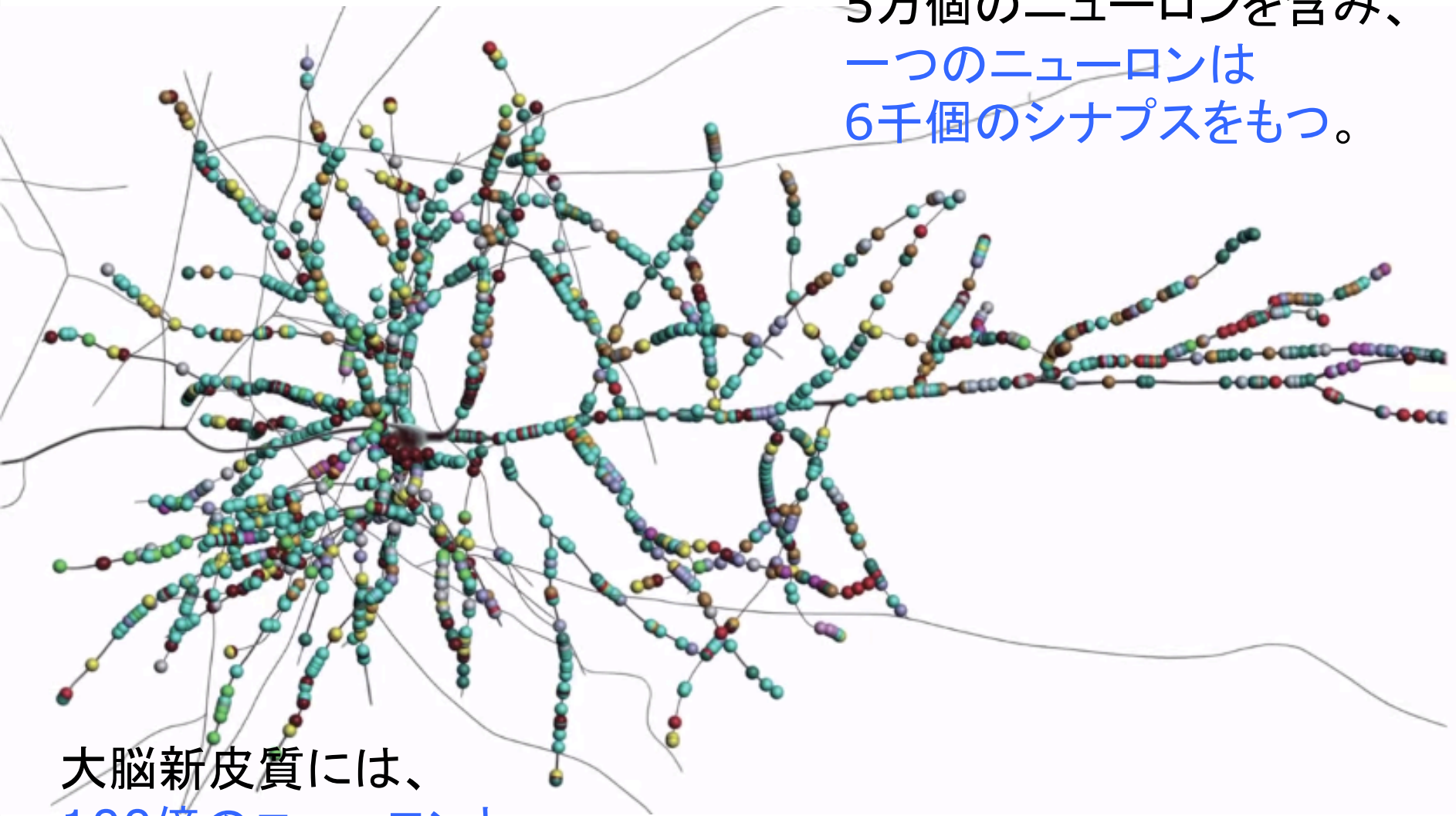
視覚と運動能力は、強く結びついている



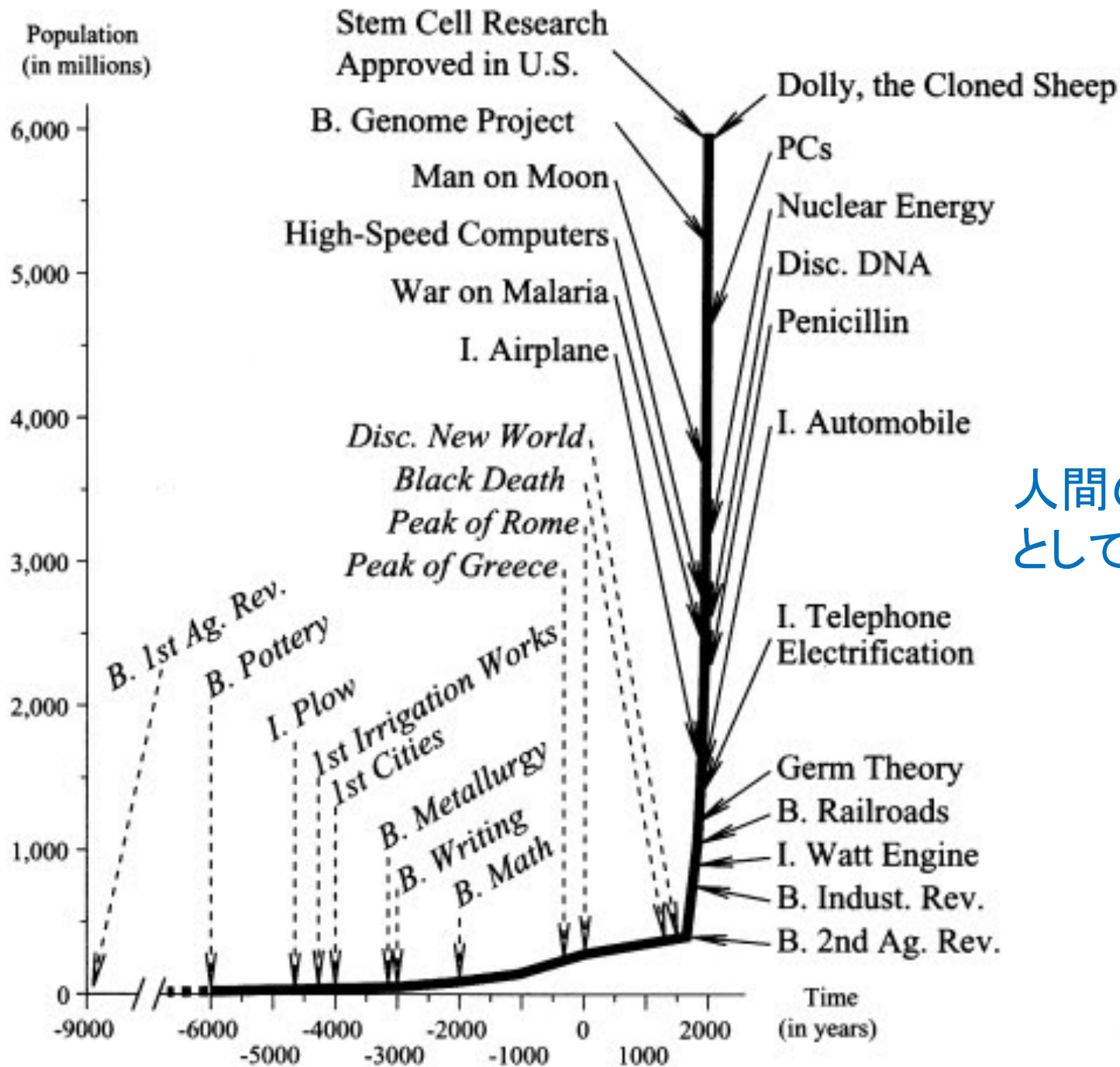
Human
Connectome
Project



脳は、1mm立方に
5万個のニューロンを含み、
一つのニューロンは
6千個のシナプスをもつ。



大脳新皮質には、
100億のニューロンと
60兆個のシナプスがある！



人間の生物種としての「成功」



人間・機械の「計算能力」を
モデルとするアプローチ

計算主義とは何か？

計算主義とは何か？

人工知能論における「計算主義」とは、人間の「知能」の本質を「計算」として抽象しようとするものである。僕は、基本的には、「計算主義」の立場に立っている。

「知能」に対する「計算主義的」なアプローチは、僕が人工知能が将来実現すべき課題として関心を持っている、人間の「言語能力」や日常的な「論理的推論能力」、抽象的な「数学的能力」、さらには「科学する能力」の機械による実現には、そうしたアプローチが不可欠だと考えている。

この時、「感情」や「意識」等々の機械による知能実現にまつわる、様々な問題を、当面は括弧に入れて保留しているのだが。

もちろん、人間の「知能」にまつわる問題を、一つの方法論で対応するのは難しい。それについては、次の「知能の階層性」で、述べようと思う。

計算主義の含意

- 「計算」を、ある入力がある出力に変換する「機能」として考えれば、同一性を残したまま、その具体的実装は捨象できる。ある計算が、そろばんの上でなされようと、スパコンの上でなされようと、はたまた、「計算」を担うのが、タンパク質だろうがシリコンだろうが、していることは「同じ」として考えることはできるのである。
- 人間の「知能」の本質を「計算」として抽象することで、その抽象の限りでは、「機械」と「人間」は等値されうる。だから、この抽象のもとでは、機械ができることは人間にもできることになり、人間ができることは機械にもできることになる。特に、後者の「人間ができることは機械にもできる」という主張は、人工知能技術の可能性を考える上では、とても強い主張になる。人工知能の可能性を考える上では、「我々自身を見よ！」という以上の強いメッセージを、僕は、思いつかない。

計算主義と複雑性の理論

「それじゃ、逆に、Googleの巨大なCPUパワーで出来ることも、人間が手計算で出来ることも同じだということ？」

そう。同じなのだ。両者が、原理的には、紙テープを1マスずつ読み書きするTuringマシンと同値だという意味では。(もともと、TuringマシンによるGoogleクラウドのシミュレーションが「多項式時間」で出来るか否かが、重要なポイントになるのだが。それについては、あとで「拡張されたチャーチ=チューリング・テーゼ」のところで述べる。)

「計算」というの能力での人間と機械の同一性が、「計算主義」の要になる概念であるなら、計算能力の比較可能性をきちんと考える必要がある。そこに、「複雑性理論」が登場する。

「複雑性理論」というのは、「計算」の「複雑さ」を階層的に整理しようとする数学の一分野である。

複雑性理論

ただ、残念なことに、「複雑性理論」、まだまだわからないことが山のようにある。有名なのは、「 $P = NP ?$ 」問題が、解けていないことである。研究の現状をまとめたサイトがここにある。

"Complexity Zoo" <https://goo.gl/eabDas>

なんと、「複雑さ」のカテゴリーが、500以上ある！

それは、「計算する知能」の構造が、予想以上に複雑だということを反映しているからだと僕は考えている。

我々は、我々自身のことをよく知らないのである。

ただ、僕は、この複雑性理論が、「計算主義」の立場から人工知能技術の可能性を考える上で、もっとも基本的な手段を提供してくれると考えている。

- 残念ながら、紙幅の関係で、「計算主義的」人工知能の実例、将来の応用分野について語ることは、今回は割愛した。
- 詳しくは、MaruLaboの次の「形式手法」ページの各エントリーを参照してほしい。
<https://www.marulabo.net/themecategory/deep-spec-2/>
- ただ、「計算主義的」人工知能技術の最も重要な応用分野は、ITの世界での開発そのものになると、筆者は考えている。
- 2019年のマルレク「IT技術とCoqの世界 --
「証明」＝「プログラム」＝「計算」の意味を考える」から、その一部を抜粋しておく。

IT技術とCoqの世界

「証明」=「プログラム」=「計算」の意味を考える



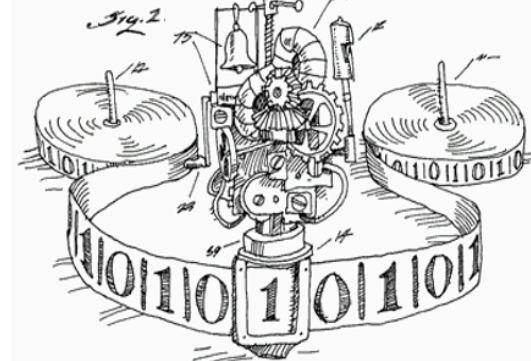
数学者

プログラム



プログラマ

計算



コンピュータ

なぜ、開発は人間によって担われているのか？

- それは、ある目的を持ったプログラムを作成する能力が人間にしかないからである。人間は、機械にはない「創造的」な能力を持っている。
- ではなぜ、人間はバグを生み出すのか？
それは、人間は「創造的」かもしれないが、「完全」ではないからである。
- ただし、人間はバグを修正できる。
機械は、現在の技術では、バグの修正はできない。

バグを修正する人間の能力はどのようなものか？

- それでは、バグを修正するのに、人間のどんな能力が使われているのだろうか？
- それはプログラムの目的に照らして、その振る舞いが正しいか否かyesかnoで判断し、正しくないなら、その原因を見つけ出すという能力である。前者と後者では、多少、難しさは違う。ただ、両者とも、「論理的」に推論する能力と言っている。
- そこで問題となっているのは、プログラムの意図を記述する「仕様」と「実装」の関係である。
- 「仕様」の「意味」を理解することが、機械には難しいことが、ここで人間が必要とされる大きな背景になる。

なぜ、開発は人間によって担われているのか？

- 開発は、単に、人間によって担われているというだけでなく、「非常にたくさん人間」によって担われている。
- それは、そうした作業を、現状の技術では、機械によっては代替できないからである。
- ただ、それは本当だろうか？

コンピュータが論理的推論能力を持てば、 開発はどう変わるか？

- 今回のセミナーは、コンピュータが論理的推論能力を持てば、開発がどのように変わるかを考えることをテーマにしている。
- 現在の開発作業の中核をなす「テスト・デバッグ・コードの見直し」というループの繰り返しをドライブしているのは、仕様との関係でのプログラムの見直しである。
- デバッグ作業でプログラマは、プログラムとにらめっこしているように見えるのだが、その時、無意識的にでも（「仕様」はドキュメントに書かれたものだけではない。「暗黙の常識」も「仕様」として機能する）、意識的にでも、常に意識しているのは、仕様とプログラムの関係である。

「仕様」と「実装」の関係は、論理的である

- 重要なことは、「仕様」と「実装」の関係は、論理的なものであるということである。
- もし、仕様が形式的に定義されているなら、ある実装がその形式的仕様を満たしているか否かは、形式的に証明可能である。
- もし、機械が論理的推論能力を持てば、そうした証明の実行によって、仕様と実装の一致をチェックするテスト・デバッグ作業に、人間は不要になる。

「実装」の正しさの鍵は、「仕様」が握っている

- 明らかに、「実装」が正しいか否かの判断の鍵は、「仕様」が握っている。
- ただ、「実装」が、どの実装でもコンピュータで受理されるプログラム言語の形式をとっているのに対して、「仕様」の記述のレベルは、まちまちである。仕様の策定者は、一般には、実装コードを書かない。
- もし、コンピュータが論理的推論を実行する能力を持ち、仕様と実装の一致を形式的に証明できるなら、ほとんど同じ技術を使って、形式的仕様から実装のプログラムが自動生成できる。
- コンピュータによる、仕様からのプログラムの自動生成は、開発の世界を大きく変える。

開発での人間の仕事はなくなるのか？

- なくなるらない。
- プログラムを作成するのに、人間の「創造性」が必要だったように、形式的な仕様を作成するには、人間の「創造的」な力がある。
- ただ、人間による誤りの混入を防ぐために、仕様作成の各ステップでの「論理性チェック」のサポートが必要になる。それは、現在のGUIを備えた統合開発環境と同じようなものだ。コンピュータが、仕様作成を支援する。

こうした見通しは実現可能か？

- 現在では、まだ、こうした環境は部分的にしか実現されていない。こうした見通しが、開発全体に広がる可能性はあるのであろうか？
- 十分にある。
 1. セキュアで堅牢なソフトウェアを提供する上で、人間から構成された開発体制の構造的複雑さは、バグの温床になる。抜本的対応がいずれ求められることになる。
 2. 開発体制の複雑さは、開発期間と開発コストの増大に結びついている。開発の多くの部分の自動化には、経済合理性がある。
 3. こうした開発を可能にする「証明支援システム」が登場し、その利用も大きく拡大している。

Part II

計算可能性理論と
計算複雑性理論

Part II

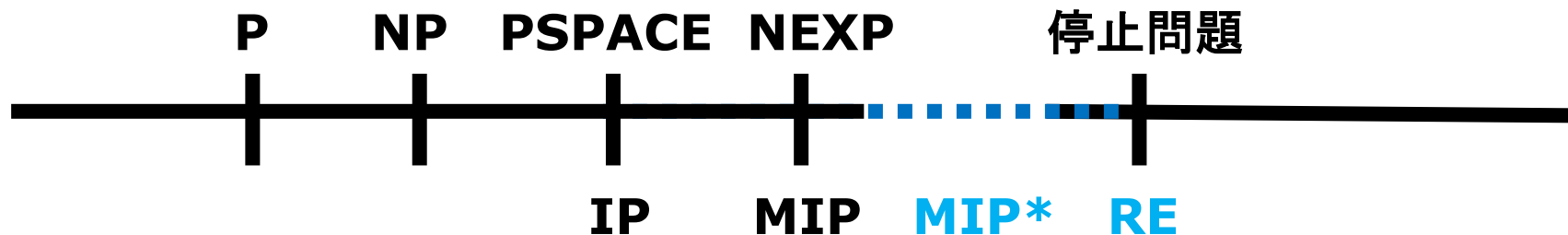
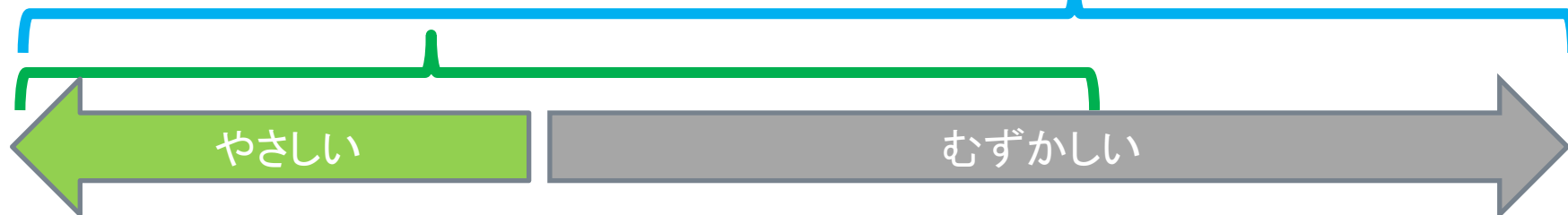
計算可能性理論と 計算複雑性理論

- 計算可能性理論 -- Church-Turing Thesis –
 - ゲーデルの不完全性定理
 - 計算可能性の定式化 -- 帰納的と帰納的可算 –
 - 計算可能だが計算が難しいものがある
- 計算複雑性理論
 - 計算複雑性理論とは何か
 - 基本的な複雑性のクラス -- P, PSPACE, EXP
 - NPとNP完全
 - 多項式時間で還元可能なNP完全問題

計算の理論と認識の難易

計算複雑性理論

計算可能性理論

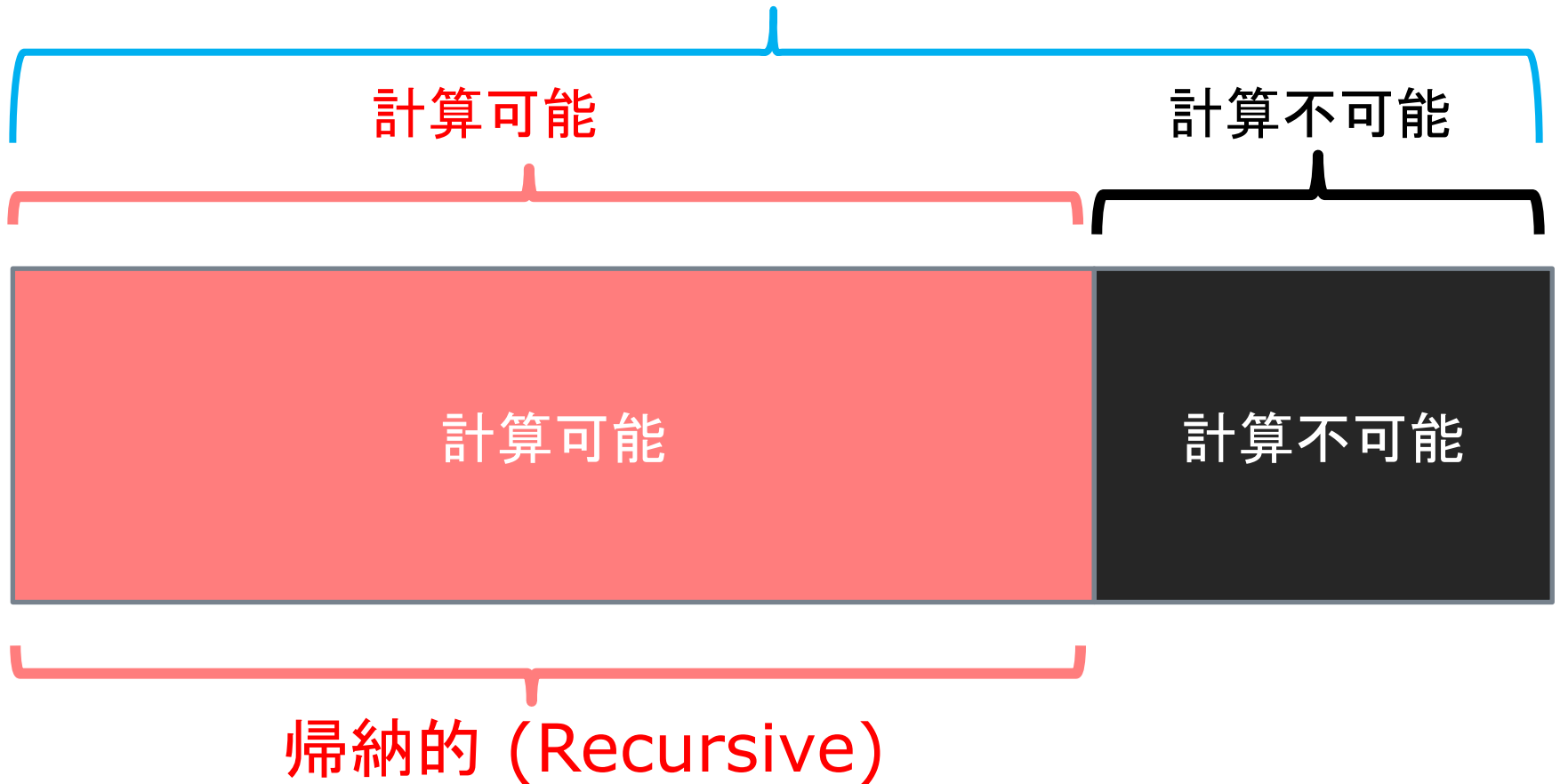


計算可能性理論

-- Church-Turing Thesis --

計算可能性理論

-- Church-Turing Thesis --



ゲーデルの不完全性定理

“On Formally Undecidable Propositions of Principia
Mathematica and Related Systems”

Kurt Gödel 英訳 by Bernard Meltzer

<https://goo.gl/PMMqYr>

ヒルベルト・プログラムと「有限の立場」

1920年、ヒルベルトは、全ての数学に確実な基礎を提供するという壮大な計画を提出する。その主要内容は、

- 数学は、形式言語で、明確なルールに従って記述されること。
- 数学的に正しい命題は、全て、この形式的システムの中で証明されること。
- この数学の形式化が矛盾を含まないことが証明されること。
その無矛盾性の証明は、有限の数学的対象に対する「有限の立場」からの証明であるべきこと。
- 「実在的な対象」について、「理想的な対象」(例えば、「非可算集合」のような)を用いて行われた証明は、それらを用いなくても証明されるべきこと。
- 全ての数学的命題の真または偽を決定する方法が存在すること。

1930年 ゲーデルの不完全性定理 (1)

ゲーデルの不完全性定理とは、次のような定理である。

「ある形式的体系 S を作ったとしよう。その体系では、初等数論を定義でき、矛盾がないとする。その時、 S に属する命題 G で、 S の中では証明も反証もできないような命題 G が存在する。」

例えば、次のような命題 G を考えてみよう。

G :「 G は証明できない。」

この命題 G が証明できたとする。それは、 G は証明できないことを証明したことになる。

この命題 G の否定が証明できたとする。それは G は証明できること、すなわち、 G は証明できないことを証明したことになる。

嘘つきのパラドックス

これは、つぎのような「嘘つきのパラドックス」と、同じような構造である。

あるクレタ人が語った。「クレタ人の言うことは全て嘘である。」

このクレタ人の言うことが嘘ではなく正しいとしよう。それは、「クレタ人の言うことは全て嘘である。」というのが正しいということなので、このクレタ人の言葉も嘘になる。

このクレタ人の言うことが嘘だとしよう。それは、彼が嘘についていないことを意味する。

1930年 ゲーデルの不完全性定理 (2)

それだけだと、そんなものかと思われるかもしれないが、ゲーデルの不完全性定理が、数学の世界に強い衝撃を与えたのは、不完全性定理の次の形である。

Fを、初等数論を含む、無矛盾の形式的体系としよう。この時、この形式的体系Fの無矛盾性は、この体系Fの中では証明出来ない。

この結果は、ヒルベルト・プログラムが、実行不可能であることを意味していた。

不完全性原理は、我々の認識の限界？

ただ、この不完全性定理は、確かに驚くべきものではあるのだが、ほとんど役に立たない定理である。

というのは、言い過ぎなのだが、この定理から、我々人間の（同時に、それは人工知能にとっての）認識能力の限界を議論するのは、あまりに、荒い議論だと僕は考えている。

一つには、不完全性定理による原理的な「限界」以外にも、形式的演繹による認識には、実質的には、様々な「限界」が存在しているからだ。我々の形式的認識の「限界」を構成しているのは、不完全性定理だけではないのだ。

（あとで見るように、複雑性理論は、もっと精緻に、その「限界」を境界づけようとしている。）

「計算とはなにか？」「証明とは何か？」 より基本的な「問い」の始まり

ゲーデルの不完全性定理は、「証明不可能性」「形式的体系の無矛盾性」についてのショッキングな定理なのだが、いかなる計算、いかなる証明が「可能」であるかについては、あまり、多くのことを我々に教えてはくれないのだ。

歴史的に見ると、不完全性定理は、証明あるいは計算の性質についての探求の中で生まれた「最後の結論」ではないのだ。事実、その逆である。1930年代に、この分野は、不完全性定理に強烈な刺激を受けて、「証明とは何か？」「計算とは何か？」という基本的な「問い」に突き進むことで発展する。

様々な「計算可能性」へのアプローチと その同値性の認識

ゲーデルの結果を受けて、「計算可能性」についての探求が一斉に始まる。

- ゲーデル＝エルブランの「帰納関数論」
 - チャーチの「ラムダ・カリキュラス」
 - チューリング＝ポストの「チューリング・マシン」
- が続々と登場する。

この時代の白眉は、「計算可能性」についての見かけは全く異なるこれらのアプローチが(もちろん、それは、ゲーデルの結果を再現できるものだ)、次々と、「同値」であることが証明されていくことだ！

様々な「計算可能性」へのアプローチと その同値性の認識



Kurt Gödel
1906-1978



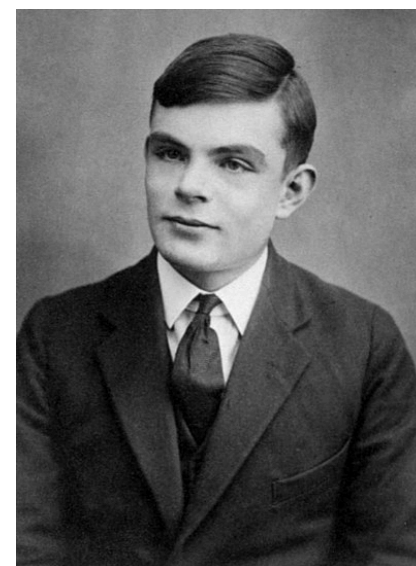
帰納関数論



Alonzo Church
1903-1995



ラムダ計算



Alan Turing
1912-1954



チューリング
マシン

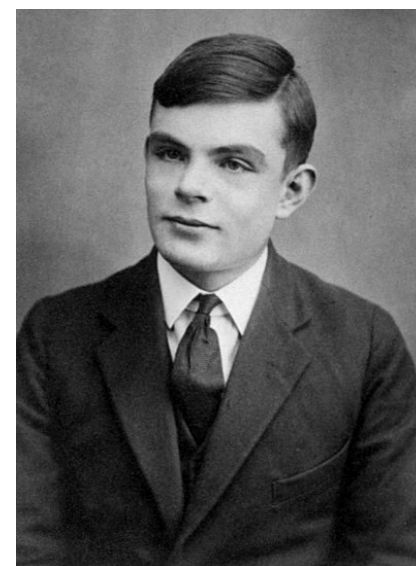
様々な「計算可能性」へのアプローチと その同値性の認識



Kurt Gödel
1906-1978



Alonzo Church
1903-1995



Alan Turing
1912-1954



帰納関数論 = ラムダ計算 = チューリング
マシン

チューリング・マシンの停止問題

- チューリング・マシンの停止問題というのは、任意のチューリング・マシーン M とそれに対する任意の入力 A が与えられた時、このチューリング・マシンは停止するか否かを求めよという問題である。
- この問題を解くチューリング・マシンは存在しない。それを「停止問題の決定不能性」という。

決定不能性の直観的説明

- 次のような関数gが、定義可能か考えてみる。

```
def g(): if halts(g): loop_forever()
```

- もし、halts(g)が真なら、すなわちgが停止するなら、loop_forever()が呼び出されて、gは停止しない。
- もし、halts(g)が偽なら、すなわちgが停止しないなら、loop_forever()を呼び出さずに、すぐにgは停止する。
- いずれにしても、矛盾するので、こうした関数は定義できない。

計算可能性の定式化

-- 帰納的と帰納的可算 --

歸納的と歸納的可算

計算可能

計算不可能



歸納的 (Recursive)



帰納的可算

recursively enumerable

- ある自然数の集合 S に対して、チューリング・マシン M が存在して、次の条件を満たす時、 S は「帰納的可算 recursively enumerable」であるという。
 - $n \in S$ なら、 M は停止して 1 を出力する。

- この定義でポイントが一つある。
 M は、 S に属するメンバーに対しては、停止して1を出力するのだが、 S に属さないメンバーに対しては、何も語っていない。
recursively enumerableなチューリング・マシン M では、全ての入力に対して M が停止するとは限らないということである。
- S に属さないメンバーに対しても、チューリング・マシン M が停止することを要求するのが、次の「帰納的 recursive」の定義である。

帰納的 recursive

- 自然数の集合 S は、次の条件を満たすチューリング・マシン M が存在する時、「帰納的 recursive な集合」と呼ばれる。
与えられた自然数 n について、
 - $n \in S$ なら、 M は停止して 1 を出力する。
 - $n \notin S$ なら、 M は停止して 0 を出力する。
- 自然数から自然数への関数 f は、次の条件を満たすチューリング・マシンが存在する時、「帰納的な関数」と呼ばれる。
 - 全ての入力 n に対して、 M は停止して、出力 $f(n)$ を返す。

計算可能な集合 / 関数は、帰納的であるという主張が、次の Church-Turing Thesis である。

チャーチ=チューリングのテーゼ

1940年代には、今日、「チャーチ=チューリングのテーゼ」と呼ばれる「計算可能性」についての認識は、確立されることになる。

「チャーチ=チューリングのテーゼ」というのは、次のような提案である。これは帰納関数論的特徴づけである。

すべての実効的に計算可能な関数(実効的に決定可能な述語)は、一般帰納的である。

チューリング・マシンによって特徴づければ、次のようになる。

計算可能な計算はすべて、ある帰納的Turingマシンによって実行される。

帰納的可算

自然数の集合 S は、次の条件を満たすチューリング・マシン M が存在する時、「**帰納的可算**」と呼ばれる

帰納的

計算
不可能

帰納的可算

- $n \in S$ なら、 M は停止して 1 を出力する。

帰納的

自然数の集合 S は、次の条件を満たすチューリング・マシン M が存在する時、「**帰納的**」と呼ばれる

帰納的

帰納的可算

計算
不可能

- $n \in S$ なら、 M は停止して 1 を出力する。
- $n \notin S$ なら、 M は停止して 0 を出力する。

Church-Turing Thesis

計算可能なものは帰納的である



計算可能だが計算が難しいものがある

帰納的(計算可能)だが 急速に増大するアッカーマン関数

次のような関数を考える。

$$\text{Ack}(m, n) = \begin{cases} n + 1, & \text{if } m = 0 \\ \text{Ack}(m - 1, 1), & \text{if } n = 0 \\ \text{Ack}(m - 1, \text{Ack}(m, n - 1)), & \text{otherwise} \end{cases}$$

この関数は、一般帰納関数として定義されているので、チャーチ=チューリングのテーゼの条件を満たしている。

ただ、 m, n の値が、少しでも大きくなると、この関数の値は、爆発的に増大して、普通のコンピュータでは、事実上、計算が不可能になる。

次ページに、 $\text{Ack}(m, n)$ の値を示す。

A(m, n) の値

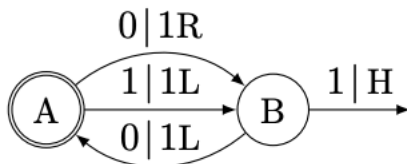
m \ n	0	1	2	3	4	n
0	1	2	3	4	5	$n + 1$
1	2	3	4	5	6	$n + 2 = 2 + (n + 3) - 3$
2	3	5	7	9	11	$2n + 3 = 2 \times (n + 3) - 3$
3	5	13	29	61	125	$2^{n+3} - 3$
4	13	65533	$2^{65536} - 3$	$2^{2^{65536}} - 3$	$A(3, A(4, 3)) = 2^{2^{2^{65536}}} - 3$	$\underbrace{2^{2^{\dots^2}}}_{n+3 \text{ twos}} - 3$
5	65533	$\underbrace{2^{2^{\dots^2}}}_{65536 \text{ twos}} - 3$	$A(4, A(5, 1))$	$A(4, A(5, 2))$	$A(4, A(5, 3))$	$A(4, A(5, n - 1))$
6	$A(5, 1)$	$A(5, A(6, 0))$	$A(5, A(6, 1))$	$A(5, A(6, 2))$	$A(5, A(6, 3))$	$A(5, A(6, n - 1))$

我々に手の届かない有限について

Busy Beaver 問題

「忙しいビーバー」問題は、状態の数がN個のTuringマシンを全部考えて、全て0で埋まっているテープから初めて、一番長いステップで停止する Turingマシンとそのステップ数を求めよという問題である。

例えば、二つの状態A,Bを持つ、次のTuringマシンは、6ステップで停止する。



initial state	...	0	0	0	0A	0	0	...
step 1	...	0	0	0	1	0B	0	...
step 2	...	0	0	0	1A	1	0	...
step 3	...	0	0	0B	1	1	0	...
step 4	...	0	0A	1	1	1	0	...
step 5	...	0	1	1B	1	1	0	...
step 6 (halt)	...	0	1	1H	1	1	0	...

2つの状態を持つTuringマシン

例：5つの状態を持つTuringマシンのプログラム

5つの状態を持つTuringマシンのプログラムの例を示す。この表の、例えば、最初の State Aの行で、On '0' の欄の 'B1R' は、「状態Aで、ヘッドが'0'の上にあるなら、状態をBに変えて、1を書き込んで、ヘッドを右(R)に移動する」という命令を表す。

State	on 0	on 1	on 0			on 1		
	0	1	Print	Move	Goto	Print	Move	Goto
A	B1R	D0L	1	right	B	0	left	D
B	C1R	F0R	1	right	C	0	right	F
C	C1L	A1L	1	left	C	1	left	A
D	E0L	H1L	0	left	E	1	left	H
E	A1L	B0R	1	left	A	0	right	B

状態が5つ

命令が10個

その最大値をBB(N)で表すことにすると、次の結果が知られている。

$$BB(1) = 1.$$

$$BB(2) = 6.$$

$$BB(3) = 21$$

$$BB(4) = 107$$

実は、まだ、誰も、BB(5)の値を知らない。

$$BB(5) \geq 47,176,870$$

というのはわかっている。(誰かが、状態の数が5つで、47,176,870ステップで停止するTuringマシンを作ったということ。ただ、それが「最大」だということとはわかっていない。)

$$BB(6) \geq 7.412 \times (10 \text{の} 36,534 \text{乗})$$

という評価が与えられている。

BB(6)が、36,534桁の数字より大きいという話をした。
2014年に、Wythagoras(怪しい名前だ)は、BB(7)について、次のような評価を示した。右側の数は、とても巨大な数だが、「有限」な数字である。

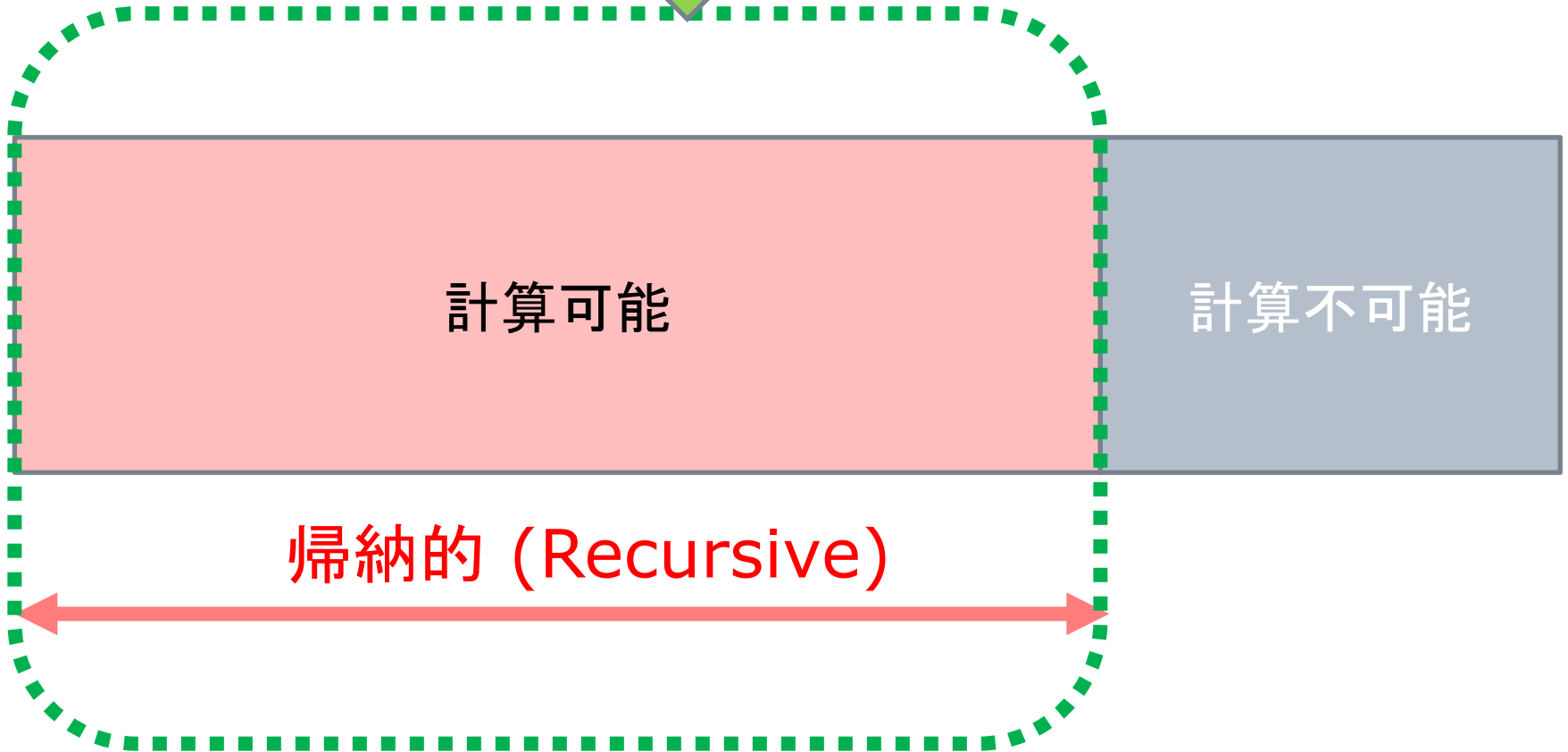
$$BB(7) > 10^{10^{10^{10^{10^7}}}}$$

2016年に、Adam Yedidia と Scott Aaronsonは、BB(7910)が、普通の集合論ZFの枠組みの中では決定できないことを証明した。

それ以来、BB(N)が計算不能になる、より小さいNの探索が、インターネット上で活発に行われ、この“logic hack”で、2018年には、BB(1919)も計算不能であることが示されたのである！

最近の研究については、Scott Aaronson “The Busy Beaver Frontier” <https://www.scottaaronson.com/papers/bb.pdf>

Church-Turing Thesis



計算可能だが計算が手に負えないものがある

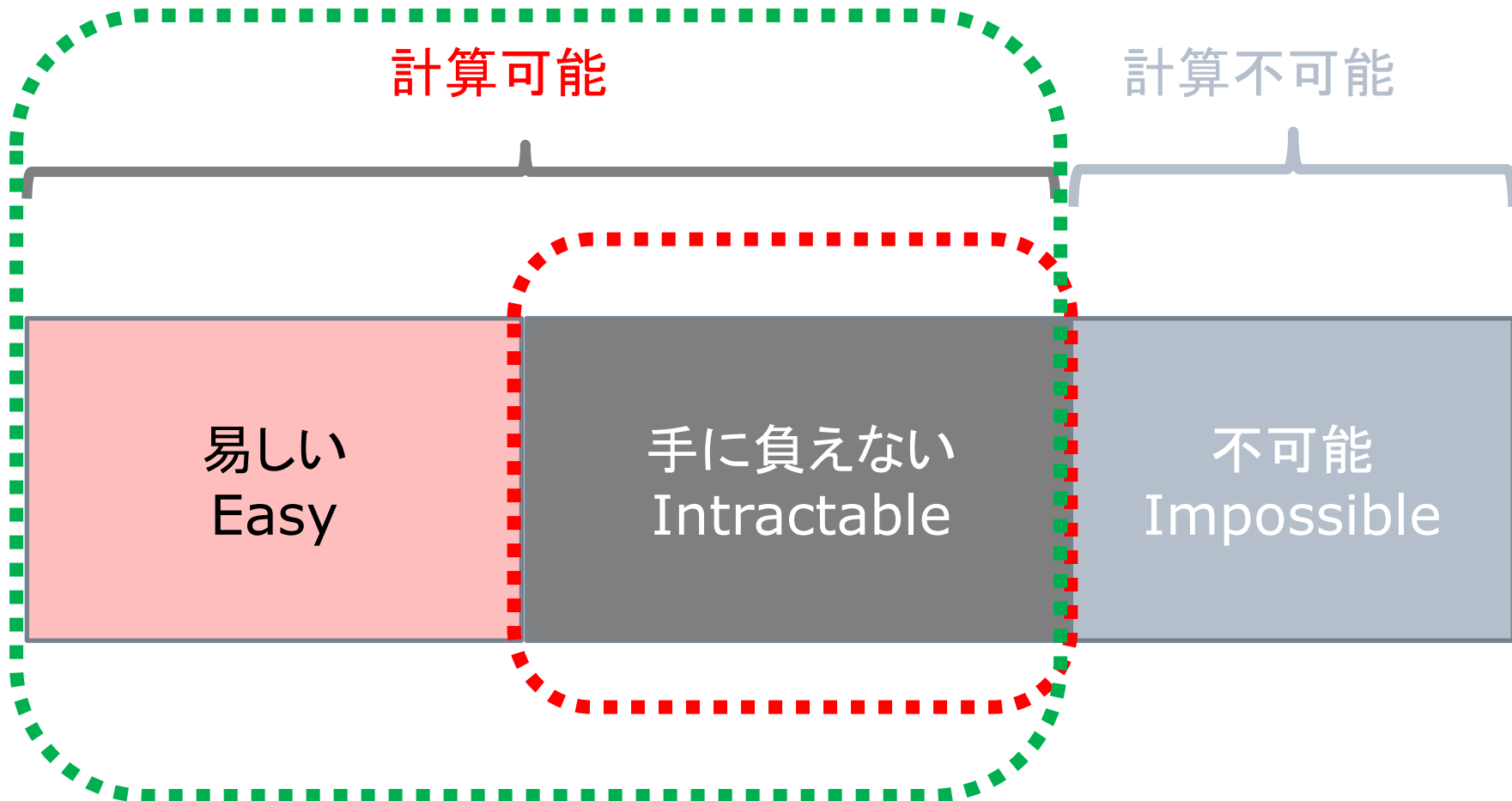
計算可能

計算不可能

易しい
Easy

手に負えない
Intractable

不可能
Impossible



20年代

ヒルベルト・
プログラム



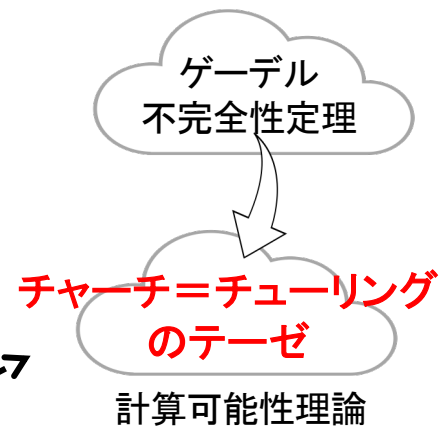
ヒルベルトは、全ての数学的命題の真または偽を決定する方法が存在すること、数学の形式化が矛盾を含まないことが証明しようとした。。

30年代



ゲーデルは、真または偽を決定する方法が存在しない命題が存在すること、また、数学の形式化が矛盾を含まないことの証明が不可能であることを示した。それはヒルベルトのプログラムが、遂行できないことを意味した。

50年代



「証明可能＝計算可能」なものには限界があることは、1950年代には、チャーチ＝チューリングのテーゼとして定式化され、一般に広く認められるようになった。ただ、それは、我々の認識の「限界」としては、非常に荒い、かつ、抽象的で原理的な限界を与えるものでしかなかった。

計算複雜性理論

計算可能性理論

計算複雑性理論

易しい
Easy

手に負えない
Intractable

不可能
Impossible

やさしい

むずかしい

P **NP** **PSPACE** **NEXP**

複雑性のクラス

計算複雑性理論とは何か

計算可能性理論の意味

理性の限界自身を数学的分析の対象とした

20世紀の初めにヒルベルトが挑戦したのは、全ての数学的問題の真偽を決定できる「純粹に機械的な手続き」をデザインすることだった。その夢は実現できなかったのだが、そうした過程が存在するのか、あるいは、存在しないとすれば何故そうなのかという疑問は残った。

そうした問いかけは、20世紀の二つの革命の登場を準備することになる。一つは、ゲーデル、チャーチ、チューリングらによる科学と哲学の分野での革命で、**理性の限界自身を数学的分析の対象とした**ことだった。もう一つの革命は、テクノロジーの分野でのコンピュータの登場だった。それは、ヒルベルトの夢を実現したわけではなかったが、地球上の多くの人々の日常生活を大きく変えた。

計算複雑性理論の始まりとその意味

ある数学的命題 S の真偽を決定する「純粹に機械的手続き」は存在しないにせよ、その S が、ある制限長さ n 以下の証明を持つかどうかを決定する手続きは存在する。単純に、長さ n 以下の証明を全て枚挙して、それが命題 S の証明になっているかをチェックすればいい。

しかし、この方法は指数関数的な時間を必要とする。 $P=NP?$ 問題は、こうした問題を解く「高速」なアルゴリズムがあるかを問うている。その意味では、 $P=NP?$ 問題は、ヒルベルトの問題提起の現代版だと考えることができる。

この問題が、明確に提起されたのは、1970年代の初めのことで、CookとLevinの仕事である。現在のようなスタイルで、 $N = NP?$ 問題を最初に定式化したのは、バークレーのStephen Cookだということ。1967年のことだ。計算複雑性理論の始まりである。

ただ、こうした問題が存在することは、50年代に、ゲーデルもナッシュも気がついていた。

基本的な複雑性のクラス

P, PSPACE, EXP

基本的な複雑性のクラス

- **Denition. P** is the class of languages $L \subset \{0, 1\}^*$ for which there exists a Turing machine M and a polynomial q so that for inputs $x \in \{0, 1\}^n$, M terminate in at most $q(n)$ steps and accepts if and only if $x \in L$.
- **Denition. PSPACE** is dened like P , except we're limited by $q(n)$ space, rather than time.
- **Denition. EXP** is dened like P , but we're limited by $2^{q(n)}$ time steps.

- **Denition. BPP** is the class of languages $L \subset \{0, 1\}^*$ for which there exists a Turing machine M and a polynomial q so that for inputs $x \in \{0, 1\}^n$, M terminate in at most $q(n)$ steps and
- if $x \in L$ then M accepts with probability $> 2/3$
 - if $x \notin L$ then M accepts with probability $< 1/3$

The constants $1/3$ and $2/3$ aren't important; we can amplify to make the success probability as high as we want by running the program a bunch of times and taking the majority result.

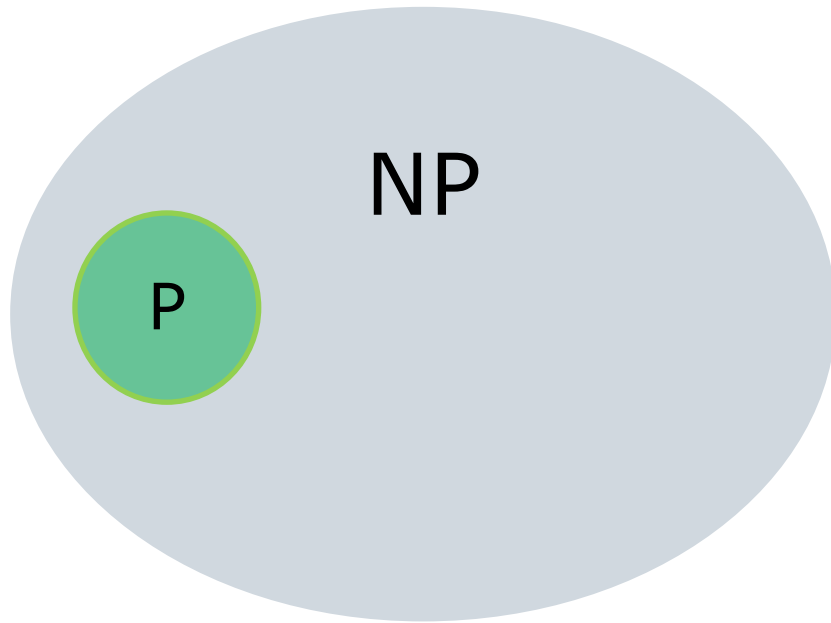
NPとNP-完全

NP問題

その問題を解くのが「多項式時間」で終わるかどうかはわからなくても、具体的な値で与えられたその問題の「答え」と言われるものが、本当に「正しい答え」であるかどうかを「多項式時間」でチェックできる問題のクラスを、NPという。素因数分解問題は、NP問題である。

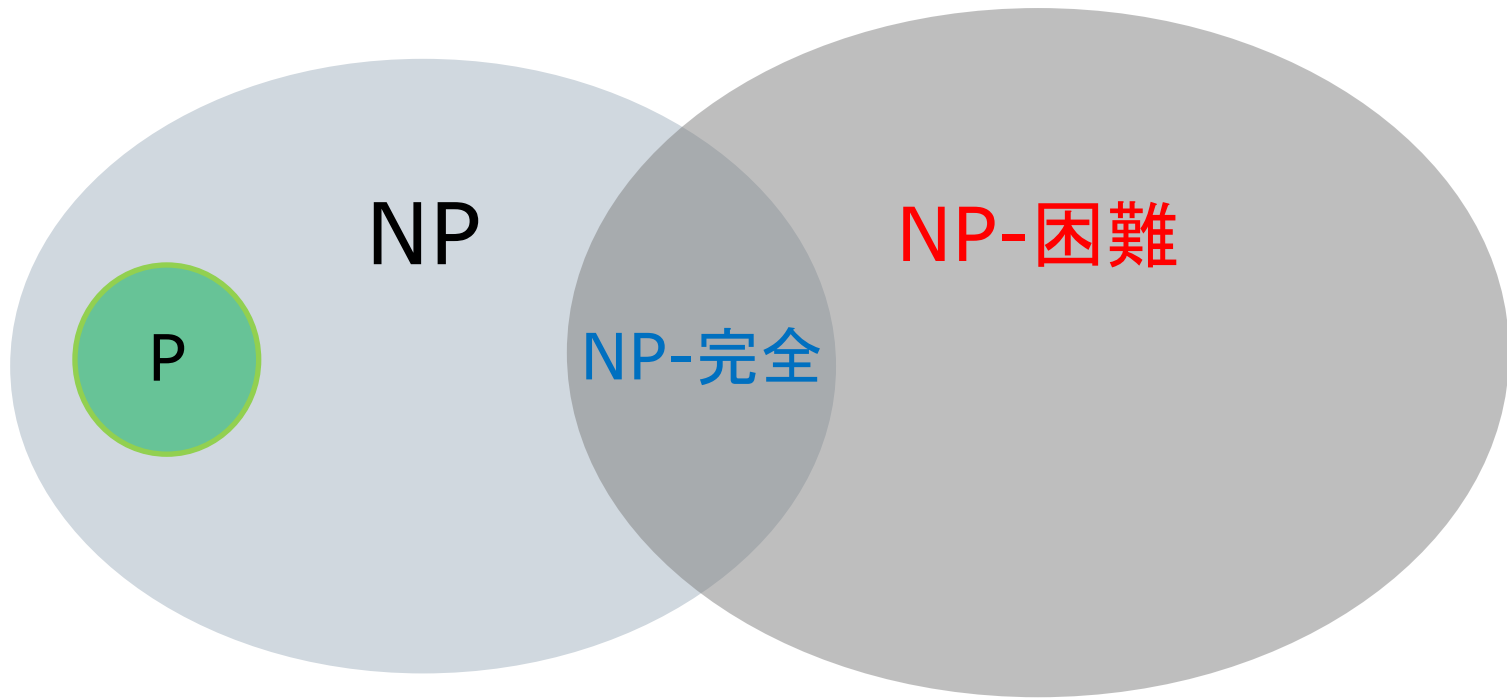
(ちなみに、「答え」(なるもの)が与えられた時、それが「間違った答え」であるかどうかを「多項式時間」でチェックできる問題のクラスを、co-NP問題という。素因数分解問題は、co-NP問題でもある。)

こうしたことや、NP問題である素因数分解問題に現れる「難しさ」と「易しさ」の「非対称性」が、現在の公開キー暗号の基礎になっていることは、多くの人が既に知っていることかもしれない。



P: 多項式時間で解ける問題

NP: 多項式時間で解けるとは限らないが、
答えが正しいかは多項式時間でチェックできる問題



P: 多項式時間で解ける問題

NP: 多項式時間で解けるとは限らないが、
答えが正しいかは多項式時間でチェックできる問題

NP-困難: 全てのNP問題が、このクラスの問題に帰着される問題
(この帰着は多項式時間で行われなければいけない)

NP-完全: NP問題で、かつ、NP-困難な問題

還元可能性 \leq_p とNP-完全

- LとMを二つの言語(問題のこと)とする。LがMに、「還元可能」というのは、多項式時間で計算可能な関数 f が存在して、

$$x \in L \Leftrightarrow f(x) \in M$$

が成り立つことをいう。この関係を $L \leq_p M$ と表す。

- 関係 \leq_p は、対称的で推移的である。
また、

$$L \leq_p M \text{ で、かつ、} M \in P \Rightarrow L \in P$$

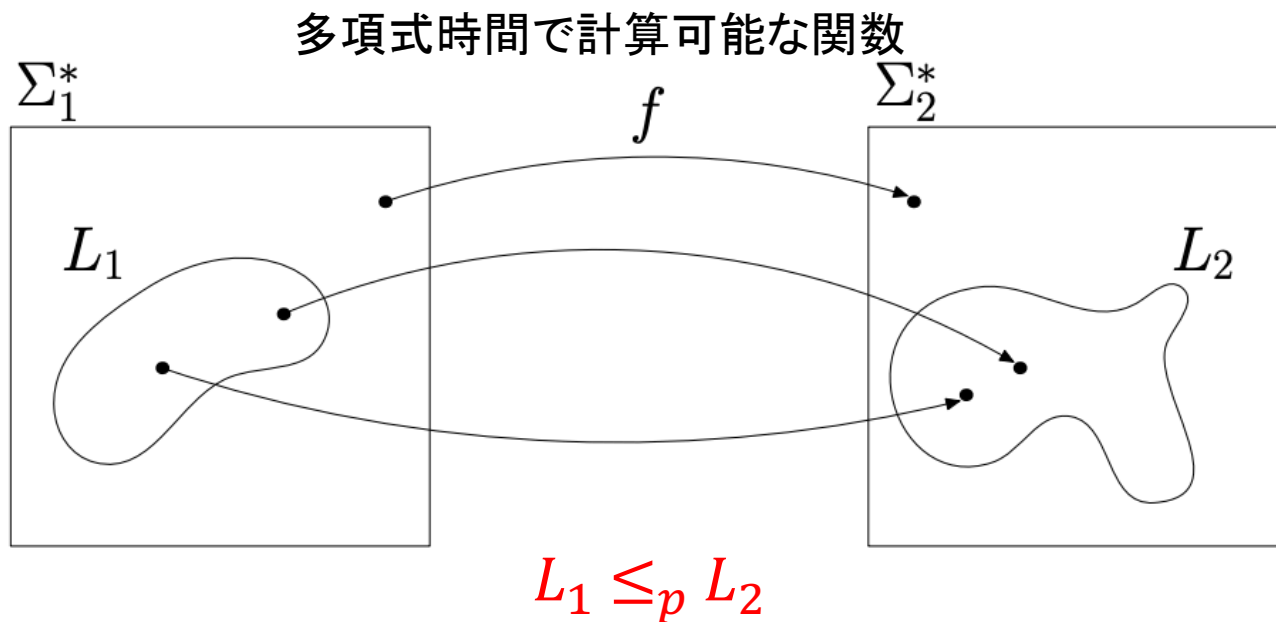
$$L \leq_p M \text{ で、かつ、} M \in NP \Rightarrow L \in NP$$

- LがNP-完全であるのは、次の二つの条件が成り立つ時

1. $L \in NP$

2. $L' \in NP$ である全ての L' について $L' \leq_p L$ (L はNP-Hard)

$$L_1 \leq_p L_2$$



- $x \in L_1 \Rightarrow f(x) \in L_2$
- $x \notin L_1 \Rightarrow f(x) \notin L_2$
- f computable in polynomial time

NP-完全問題

- 1971年、Cookは、充足可能性問題が、NP-完全であることを証明した。ここに初めて、NP-完全というコンセプトが登場する。(同じ頃、ソヴィエトの数学者Levinも同じ結果を得ている) 現代の複雑性理論は、この発見から始まる。
- 1972年、Karpは、ある問題がNP-完全であることを証明する標準的な「還元」の手法を開発し、多くの問題がNP-完全であることを示した(「Karpの21の問題」)。
- このNP-完全問題のリストは、現在では数百の問題をカバーして拡大している。(例えば、“List of NP-complete problems” <https://goo.gl/XUSvRz>)
- これらの問題の「手に負えなさ: “Intractability ”」は、例外的なものではなく、一般的なものである。
- Cookは1982年、Karpは1985年、Turing賞を受賞した。

Cook-Levin Theorem

SATはNP完全である



SAT問題

$$\begin{aligned} & (l_1 \vee l_2 \vee x_2) \wedge \\ & (\neg x_2 \vee l_3 \vee x_3) \wedge \\ & (\neg x_3 \vee l_4 \vee x_4) \wedge \cdots \wedge \\ & (\neg x_{n-3} \vee l_{n-2} \vee x_{n-2}) \wedge \\ & (\neg x_{n-2} \vee l_{n-1} \vee l_n) \end{aligned}$$

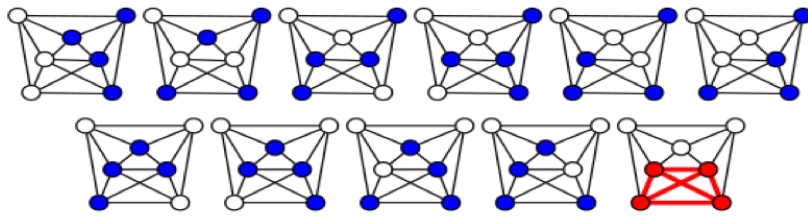
3-SAT

多項式時間で還元可能なNP完全問題

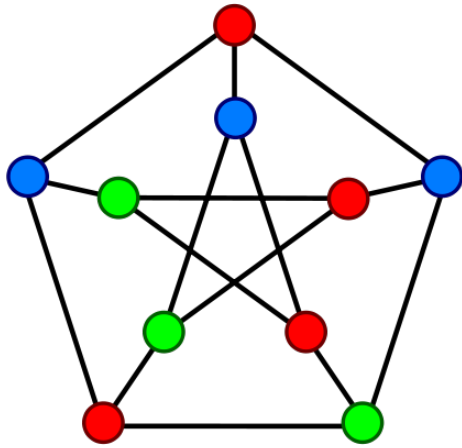
"Reducibility Among Combinatorial Problems"

Richard M Karp 1972

<https://people.eecs.berkeley.edu/~luca/cs172/karp.pdf>



Clique



Graph Coloring

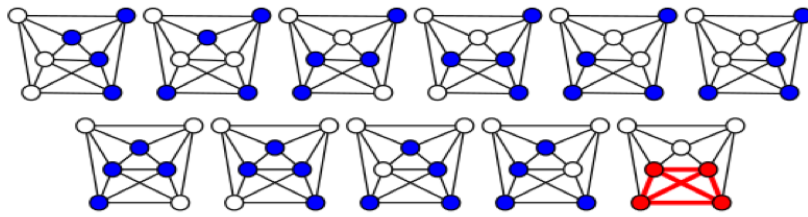
$$\begin{aligned}
 &(l_1 \vee l_2 \vee x_2) \wedge \\
 &(\neg x_2 \vee l_3 \vee x_3) \wedge \\
 &(\neg x_3 \vee l_4 \vee x_4) \wedge \cdots \wedge \\
 &(\neg x_{n-3} \vee l_{n-2} \vee x_{n-2}) \wedge \\
 &(\neg x_{n-2} \vee l_{n-1} \vee l_n)
 \end{aligned}$$

3-SAT

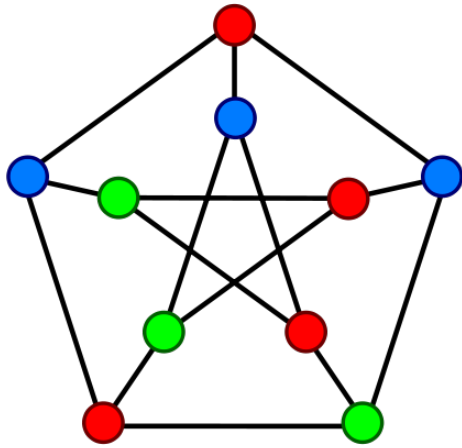


Hamilton Path

様々なNP-完全問題



Clique



Graph Coloring

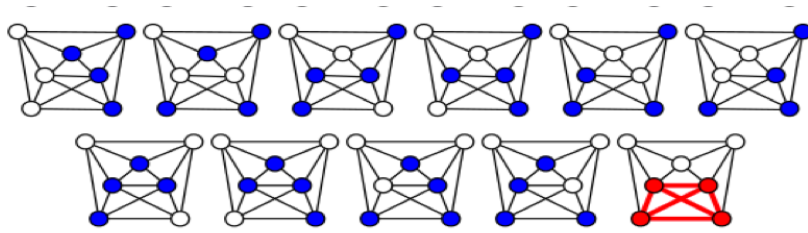
$$\begin{aligned}
 &(l_1 \vee l_2 \vee x_2) \wedge \\
 &(\neg x_2 \vee l_3 \vee x_3) \wedge \\
 &(\neg x_3 \vee l_4 \vee x_4) \wedge \cdots \wedge \\
 &(\neg x_{n-3} \vee l_{n-2} \vee x_{n-2}) \wedge \\
 &(\neg x_{n-2} \vee l_{n-1} \vee l_n)
 \end{aligned}$$

3-SAT

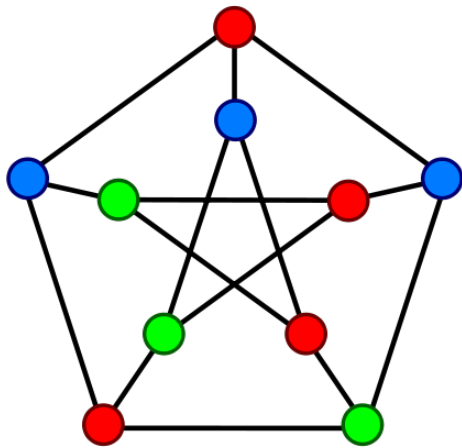


Hamilton Path

これらの問題は、解くのが難しい。もちろん、力まかせに、総当たりでやれば解けるのだが、それには指数関数的な膨大な時間がかかる。



Clique



Graph Coloring

$$\begin{aligned}
 &(l_1 \vee l_2 \vee x_2) \wedge \\
 &(\neg x_2 \vee l_3 \vee x_3) \wedge \\
 &(\neg x_3 \vee l_4 \vee x_4) \wedge \cdots \wedge \\
 &(\neg x_{n-3} \vee l_{n-2} \vee x_{n-2}) \wedge \\
 &(\neg x_{n-2} \vee l_{n-1} \vee l_n)
 \end{aligned}$$

3-SAT



Hamilton Path

しかも、この見かけは全く異なる、「現実的には手に負えない問題」は、すべて同一のアルゴリズムで、解くことができるのだ！

Karp's 21 NP-complete problems

- **Satisfiability**: the boolean satisfiability problem for formulas in conjunctive normal form (often referred to as SAT)
- **0–1 integer programming** (A variation in which only the restrictions must be satisfied, with no optimization)
- **Clique** (see also independent set problem)
 - **Set packing**
 - **Vertex cover**
 - **Set covering**
 - **Feedback node set**
 - **Feedback arc set**
 - **Directed Hamilton circuit** (Karp's name, now usually called **Directed Hamiltonian cycle**)
 - **Undirected Hamilton circuit** (Karp's name, now usually called **Undirected Hamiltonian cycle**)

Karp's 21 NP-complete problems

- **Satisfiability with at most 3 literals per clause** (equivalent to 3-SAT)
- **Chromatic number** (also called the Graph Coloring Problem)
 - **Clique cover**
 - **Exact cover**
 - **Hitting set**
 - **Steiner tree**
 - **3-dimensional matching**
 - **Knapsack** (Karp's definition of Knapsack is closer to Subset sum)
 - **Job sequencing**
 - **Partition**
 - **Max cut**

<https://goo.gl/56qLWk>

NP-hard

NP-complete

NP

P

Hamilton cycle
Steiner tree
Graph 3-coloring
Satisfiability
Maximum clique
...

Graph connectivity
Primality testing
Matrix determinant
Linear programming
...

Matrix permanent
Halting problem
...

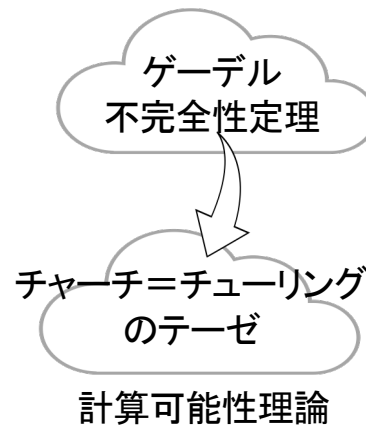
Factoring
Graph isomorphism
...



70年代



P=NP?
計算複雑性理論



クック、レビン、カープらは、計算可能だが実際には手に負えない計算がある領域を精緻に分類しようとし、計算複雑性理論が生まれる。

「難しい計算」が「易しい計算」には還元できないことを主張する「P = NP ? 問題」は、現在も未解決である。

Part III

量子コンピュータと計算科学
-- 量子複雑性理論 --

Part III

量子コンピュータと計算科学

-- 量子複雑性理論 --

□ 計算可能性理論の発展

- Church-Turing-Deutsch Thesis
- 拡張されたChurch-Turing Thesis

□ 量子複雑性理論

- Shorの発見
- 「量子超越性」の実証
- 再び、Church-Turing Thesis について

□ $MIP^* = RE$ 定理

- Interactive Proof
- 人工知能論とInteractive Proof

計算可能性理論の発展

「計算可能性理論」が、大きな転機を迎えるのは、1980年代に入ってからだ。

ファインマンが、コンピュータでは量子力学の法則に従う自然のシミュレートができないことに気づく。彼は、自然のシミュレートが可能なコンピュータは、量子力学の法則に従ったコンピュータでなければならないと主張する。

この指摘が、「量子コンピュータ」研究の始まりである。

数学的体系と同様に、自然もまた、我々の認識の対象である。数学だけでなく、物理学もまた、我々の認識の可能性と限界について、強い関心を持っているのだ。

ドイッチェは、ファインマンの考えを受けて、チューリングマシンの量子版を構成し、こうしたチューリング・マシンで計算可能なものが計算可能であるとする。これを、計算可能性についての「チャーチ=チューリング=ドイッチェのテーゼ」という。

一見すると、同じような定式化に思えるのだが、本質的な違いがあるのだ。

「チャーチ=チューリングのテーゼ」は、抽象的・形式的・数学的な「計算可能性」の定義についての提言なのだが、「チャーチ=チューリング=ドイッチェのテーゼ」は、「計算可能性」が、実在的・物理的に定義されねばならないと主張しているのだ。

「チャーチ=チューリングのテーゼ」は、いわば、遠くの雲の上に抽象的に存在する原理だったが、「チャーチ=チューリング=ドイッチェのテーゼ」は、地上に降りた現実の物理的原理だ。

重要なことは、こうした「計算可能性」概念の「物理化」の背景にある思想である。それは、情報過程が、けっして抽象的なものではなく、物理的なものに支えられた物理過程に他ならないという考えである。

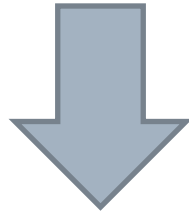
ただ、80年代は、まだ、量子コンピュータは概念としてしか存在していなかった。「計算可能性」の物理化という画期的な変化も、まだまだ、抽象的な議論だった。

こうした動きのすぐれた解説は、彼の死後に発刊されたファインマンの最後の著作である "Feynman Lectures on Computation" <https://goo.gl/PyqgyT> の第5章 "Reversible Computation and the Thermodynamics of Computing" と第6章 "Quantum Mechanical Computers" をみるのがいい。30年前のものだが、少しも古くない。

ただ、「チャーチ=チューリング=ドイッチェのテーゼ」は、その認識論的意味を考えると、本当は、なかなか手強いものだ。その辺の議論は、あとで振り返ることがあると思うが、ニールセンの次の投稿を見て欲しい。 "Interesting problems: The Church-Turing-Deutsch Principle" <https://goo.gl/Q6EiYi>

Church-Turing-Deutsch Thesis

- **Church-Turing Thesis** (Original Form):
すべての計算は、あるTuringマシンによって実行される。

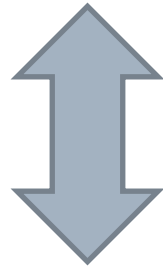


- **Church-Turing-Deutsch Thesis:**
物理システムによって実行されるすべての計算は、あるTuringマシンによって実行される。

拡張されたChurch-Turing Thesis

□ The extended CT thesis:

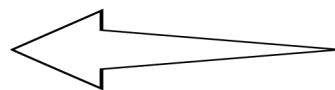
Turingマシンで、多項式時間で効率的に実行できない計算は、いかなる物理システムによっても実行できない。



物理システムによって実行されるすべての計算は、あるTuringマシンによって**多項式時間**で実行される。

80年代

P=NP?
計算複雑性理論



ゲーデル
不完全性定理



チャーチ=チューリング
のテーゼ
計算可能性理論

計算可能性概念の「物理化」
情報過程 = 物理過程



チャーチ=チューリング=ドイッチェ
のテーゼ



物理的計算可能性



ドイッチェは、ファインマンの自然をシミュレートする量子コンピュータというアイデアに刺激を受けて、計算は、物理的な過程として実現されると主張して、チャーチ=チューリングのテーゼを拡大した。80年代は、まだ、量子コンピュータは概念としてしか存在していなかった。

量子複雜性理論

量子複雑性理論

- 1993年に、Bernstein と Vazirani は、これまでのTuringマシンの拡大である「量子Turingマシン」を新しく定義して、その上で複雑性理論を展開した。
- ここから始まったこの複雑性理論の新しい分野を「量子複雑性理論」と呼ぶ。量子複雑性理論は、現在の複雑性理論の中心分野である。

BQP

- 量子複雑性理論で最も基本的なクラスは、BQPである。それは、従来の複雑性理論での多項式時間で決定可能な複雑性のクラス P に相当するものである。
- ただし、量子Turingマシンの特性として、その出力は古典的なTuringマシンのように常に確定した値を返すのではなく、確率分布として与えられる。
- BQPは、“Bounded error, Quantum, Polynomial time”の略である。この Bounded error は、このマシンの出力の「誤り」が一定の確率（一般には $1/3$ を使う）以下であることを表している。
- その点では、BQPは古典的な複雑性理論でのBPP (bounded-error probabilistic polynomial time)によく似ている。

BPPの定義

- **Denition. BPP** is the class of languages $L \subset \{0, 1\}^*$ for which there exists a Turing machine M and a polynomial q so that for inputs $x \in \{0, 1\}^n$, M terminate in at most $q(n)$ steps and
- if $x \in L$ then M accepts with probability $> 2/3$
 - if $x \notin L$ then M accepts with probability $< 1/3$

The constants $1/3$ and $2/3$ aren't important; we can amplify to make the success probability as high as we want by running the program a bunch of times and taking the majority result.

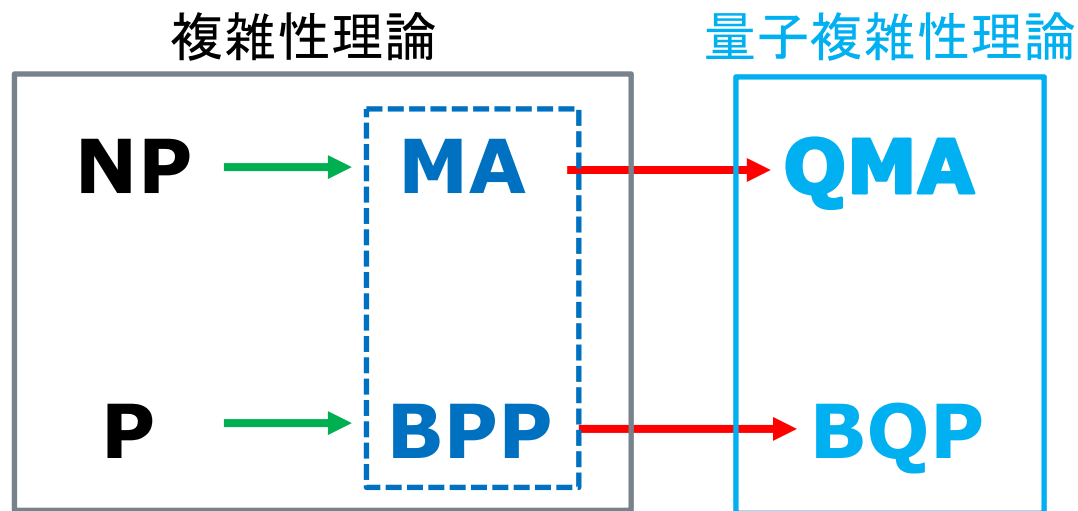
BQPの定義

- **Definition. BQP** is the class of languages $L \subseteq \{0, 1\}^*$ for which there exists a uniform family of polynomial-size quantum circuits $\{C_n\}$ over some basis of universal gates and a polynomial q so that for all n and inputs $x \in \{0, 1\}^n$
- if $x \in L$ then $C_n(|x\rangle |0\rangle^{\otimes q(n)})$ accepts with probability $> 2/3$
 - if $x \notin L$ then $C_n(|x\rangle |0\rangle^{\otimes q(n)})$ accepts with probability $< 1/3$

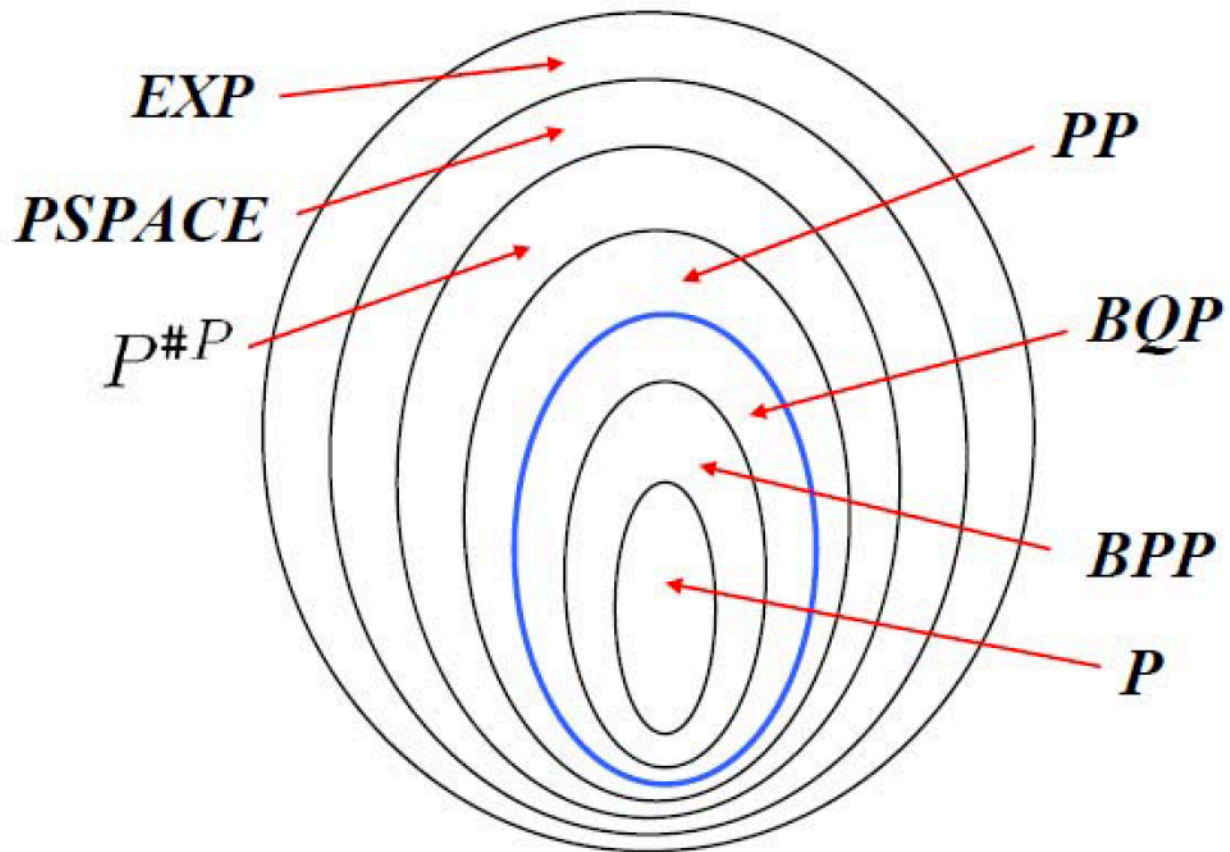
Since circuits have to pre-specify the input size, so we need a circuit for each input size n . By uniform, we mean that there is a classically efficient algorithm to produce C_n given n .

BQPとQMA

- BQPとならんで、量子複雑性理論で基本的なクラスは、QMA (Quantum Merlin Arthur)である。それは、古典的な複雑性理論のNPクラスの量子版である。(正確にいうと、NPクラスの確率版であるMAクラスの量子版である)



Basic properties of BQP



PSPACE-complete problems

□ Logic

Quantified boolean formulas · First-order logic of equality · Provability in intuitionistic propositional logic · Satisfaction in modal logic S4 · First-order theory of the natural numbers under the successor operation · First-order theory of the natural numbers under the standard order · First-order theory of the integers under the standard order · First-order theory of well-ordered sets · First-order theory of binary strings under lexicographic ordering · First-order theory of a finite Boolean algebra · Stochastic satisfiability · Linear temporal logic satisfiability and model checking

□ Lambda calculus

<https://goo.gl/xZJYF5> <https://goo.gl/hXveKm>

PSPACE-complete problems

□ Formal languages

Word problem for [context-sensitive language](#) · Intersection emptiness for an unbounded number of [regular languages](#) · [Regular expression star freeness](#) · [Equivalence problem](#) for [regular expressions](#) · [Emptiness problem](#) for [regular expressions](#) with intersection. · [Equivalence problem](#) for star-free [regular expressions](#) with squaring.

□ Quantified Boolean formulas

□ Puzzles and games

Shorの発見

“Polynomial-Time Algorithms for Prime Factorization
and Discrete Logarithms on a Quantum Computer”

1997年

<https://goo.gl/kf4ScC>

1994年、ショアは驚くべき発見をする。(論文は、1997年)

量子コンピュータでは、古典的なコンピュータでは指数関数的時間を要する素因数分解が、多項式時間で解けるという発見である。現代のセキュリティ技術の根幹部分が、RSA暗号のように素因数分解の困難性に基礎を置いている事情から、この「ショアのアルゴリズム」の発見は、コンピュータ・サイエンスの枠を超えた、社会的と言っていい大きな反響を呼び起こした。

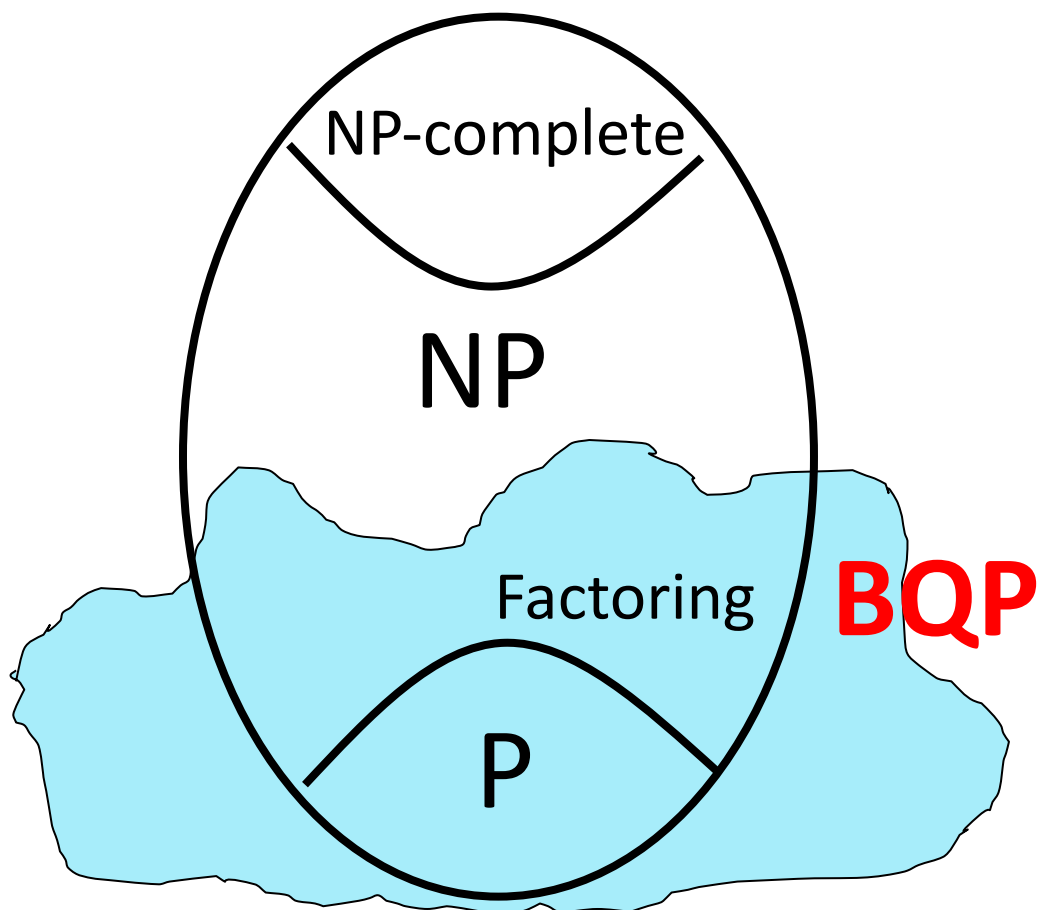
*If computers that you build are quantum,
Then spies everywhere will all want 'em.
Our codes will all fail,
And they'll read our email,
Till we get crypto that's quantum, and daunt 'em.*

Abstract

A digital computer is generally believed to be an efficient universal computing device; that is, it is believed able to simulate any physical computing device with an increase in computation time of at most a polynomial factor. This may not be true when quantum mechanics is taken into consideration. This paper considers factoring integers and finding discrete logarithms, two problems which are generally thought to be hard on a classical computer and have been used as the basis of several proposed cryptosystems. Efficient randomized algorithms are given for these two problems on a hypothetical quantum computer. These algorithms take a number of steps polynomial in the input size, e.g., the number of digits of the integer to be factored.

BQP (Bounded-Error Quantum Polynomial-Time): The class of problems solvable efficiently by a quantum computer, defined by Bernstein and Vazirani in 1993

Shor 1994: Factoring integers is in **BQP**



Example Problems



$n \times n$ chess

$n \times n$ Go

Box packing

Map coloring

Traveling salesman

$n \times n$ Sudoku

Graph isomorphism

Factoring

Discrete logarithm

Graph connectivity

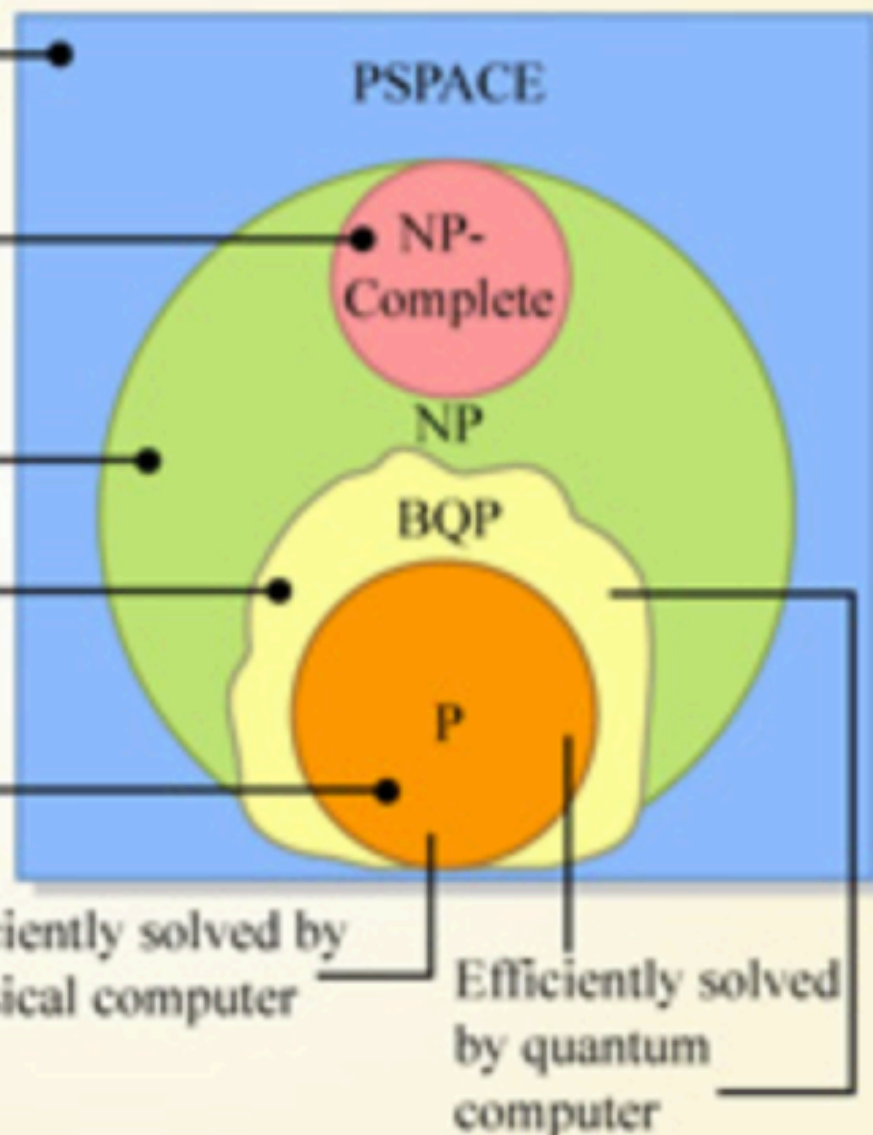
Testing if a number

is a prime

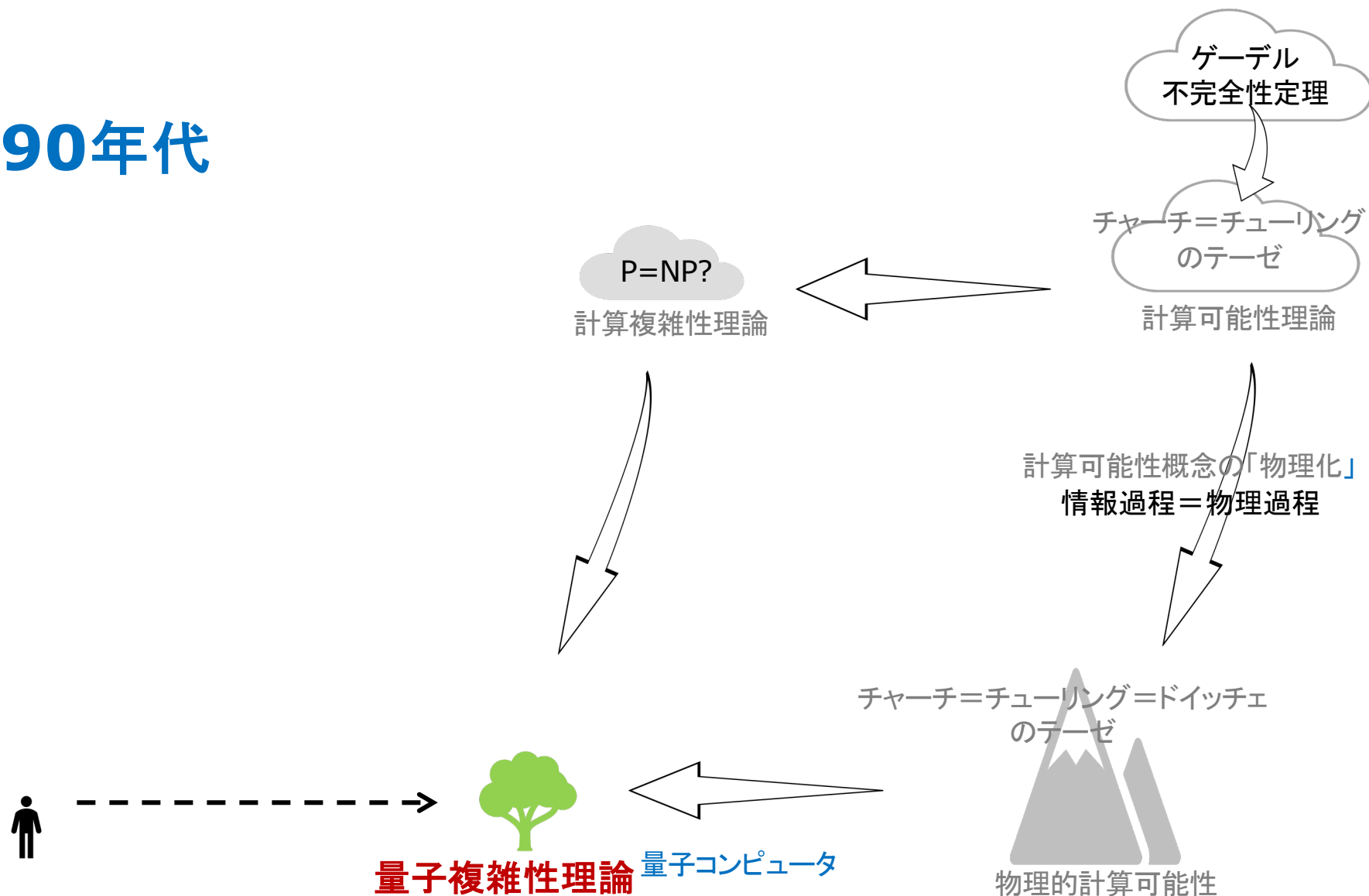
Matchmaking

Efficiently solved by
classical computer

Efficiently solved
by quantum
computer



90年代



ベルンシュタインとバジラーニによって、量子複雑性の理論が登場し、BQPという新しい複雑性の概念が提案される。広い関心をつつめたショアのアルゴリズムは、このBQPに属するものだった。

90年代

ゲーデル
不完全性定理

チャーチ=チューリング
のテーゼ

計算可能性理論

P=NP?
計算複雑性理論

その一方で、物理的な世界で計算されるものは、Turingマシンによって多項式時間で計算されるものであるという「拡大されたチャーチ=チューリングのテーゼ」が認められるようになる。

計算可能性概念の「物理化」
情報過程 = 物理過程

チャーチ=チューリング=ドイッチェ
のテーゼ

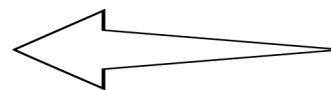
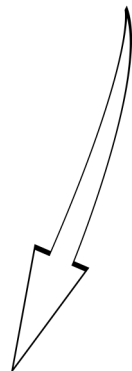
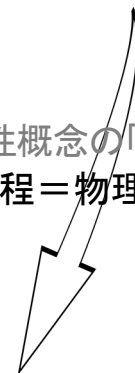
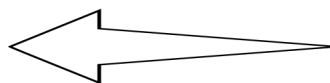
物理的計算可能性

拡大されたチャーチ=チューリング
のテーゼ



量子複雑性理論

量子コンピュータ



「量子超越性」の実証

量子優越性とは何か？

- 「量子優越性」という言葉は、2012年にプレスキルが論文 “Quantum computing and the entanglement frontier” で提唱した造語である。
- 彼はこう言っている。
Classical systems cannot in general simulate quantum systems efficiently.
(古典システムは、一般には、量子システムを効率的にはシミュレートできない)

Quantum computing and the entanglement frontier

John Preskill 2012年

<https://arxiv.org/pdf/1203.5813.pdf>

Googleによる「量子超越性」の達成

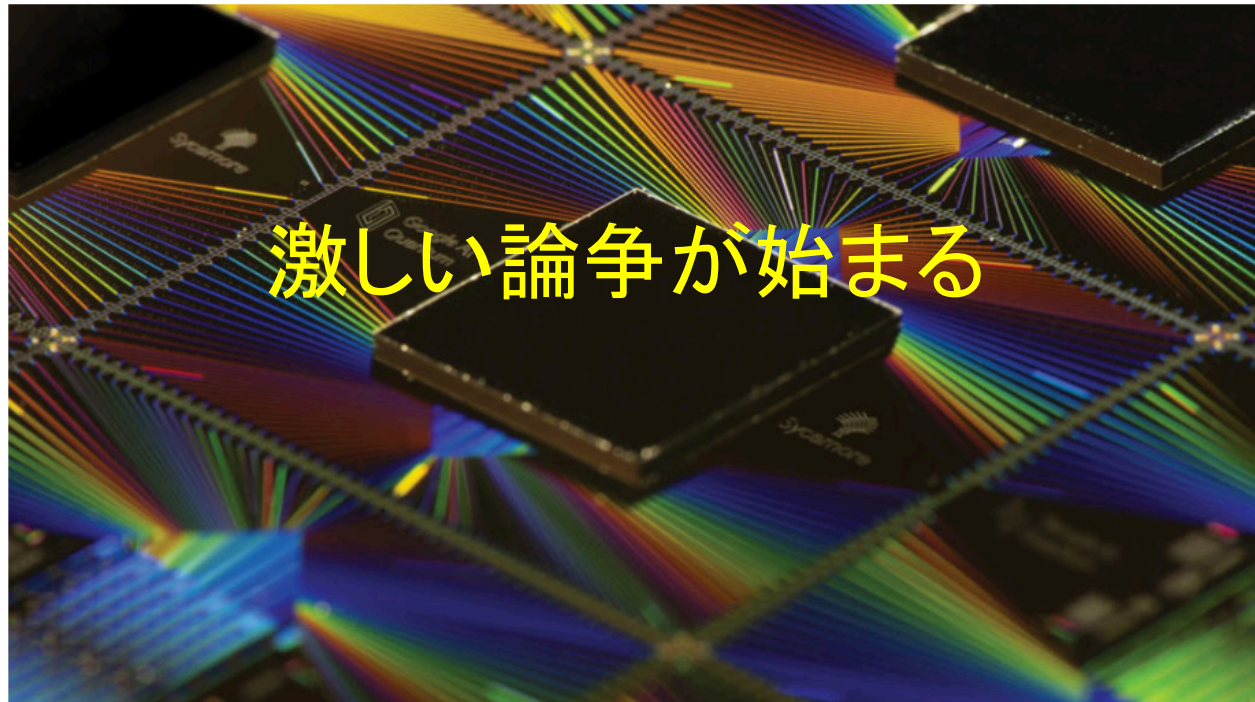
- 2019年10月、GoogleはNature誌に論文を投稿し、同社の量子コンピュータが、「量子超越性」を達成したと発表した。
- この発表は、BitCoin相場を暴落させ、また実験の評価をめぐって、激しい論争が巻き起こった。
- このあたりの問題については、今年2月のマルレク「量子コンピュータの現在 — 量子優越性のマイルストーンの達成 —」<https://www.marulabo.net/docs/q-supremacy/> を参照されたい。

YEAR IN REVIEW QUANTUM PHYSICS

Google claimed quantum supremacy in 2019 — and sparked controversy



Competitors questioned whether the milestone had truly been achieved



Google's quantum computer Sycamore performed a calculation that would take thousands of years with a classical supercomputer, researchers claimed in 2019. An array of quantum computer chips is shown.

GOOGLE

<http://bit.ly/31MgLV6>

量子超越性は、 「拡張されたChurch-Turing Thesis」 が成り立たないことを示す

- Scott Aaronsonは、Googleによる量子超越性の達成を、次のような定式化での「拡張されたChurch-Turing Thesis」が成り立たないことを実験的に示したものと評価した。
- なぜなら、Googleの実験は、量子コンピュータが古典的コンピュータ(その計算能力はTuringマシンに等しい)を超える計算能力を持つことを示したからだ。

拡張された**Church-Turing Thesis**

物理的な世界で計算されるものは、
Turingマシンによって多項式時間で
計算されるものである

量子超越性は、 「拡張されたChurch-Turing Thesis」 が成り立たないことを示す

- Scott Aaronsonは、Googleによる量子超越性の達成を、次のような定式化での「拡張されたChurch-Turing Thesis」が成り立たないことを実験的に示したものと評価した。
- なぜなら、Googleの実験は、量子コンピュータが古典的コンピュータ(その計算能力はTuringマシンに等しい)を超える計算能力を持つことを示したからだ。

~~拡張されたChurch-Turing Thesis~~

~~物理的な世界で計算されるものは、
Turingマシンで多項式時間で
計算されるものである~~

「拡張されたChurch-Turing Thesis」 が成り立たないことの人工知能論への影響

- 「拡張されたChurch-Turing Thesis」が成り立たないという発見は、人工知能論に大きな意味を持ちうる。
- 「計算主義」的人工知能論は、原理的には、人間の知能が機械の計算能力に還元可能だと考える。その限りでは、機械と人間の能力は同等である。
- 量子コンピュータが、計算のモデルとしてのTuringマシンを上回る計算能力を持つのなら、機械としての量子マシンは、人間の知能を超える能力を持つことになる。
- 誤解してはいけないのは、量子コンピュータが NP-完全問題や停止問題を解く能力を持つ訳ではないということである。それでも、複雑性理論的には、BQPは「Turingマシンによって多項式時間で計算されるもの」より、強力な計算能力なのである。それは、人間には真似できないものだ。

2019年

ゲーデル
不完全性定理

チャーチ=チューリング
のテーゼ
計算可能性理論

P=NP?
計算複雑性理論

Googleの実験は、量子コンピュータが古典的コンピュータ(その計算能力はTuringマシンに等しい)を超える計算能力を持つことを示し、拡大されたチャーチ=チューリングのテーゼが成り立たないことを示した。

計算可能性概念の「物理化」
情報過程 = 物理過程



量子超越性

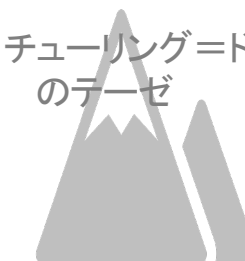


量子複雑性理論

~~拡大されたチャーチ=チューリング
のテーゼ~~

量子コンピュータ

チャーチ=チューリング=ドイッチェ
のテーゼ



物理的計算可能性

再び、 Church-Turing Thesis について

ここでは、一部重複するが、Church-Turing Thesisの
バリエーションについてまとめておこう

Church-Turing Thesis

□ Church-Turing Thesis :

すべての計算可能な計算は、あるTuringマシンによって実行される。

□ ここでの「計算可能性」は、数学的に構成されたTuringマシンと対応づけられた数学的に定義されたものである。

Church-Turing-Deutsch Thesis

□ Church-Turing-Deutsch Thesis:

物理システムによって実行されるすべての計算可能な計算は、あるTuringマシンによって実行される。

□ ここでの計算は、抽象化された数学的な計算ではなく、物理的システムによって実行される物理的なものである。

□ この命題を言い換えれば、
Turingマシンによって実行できない計算は、物理的システムでは実行できない

ということになるのだが、この命題は物理的な法則の限界として計算可能性について語っている。

Turingマシンによって実行できない計算は、物理法則を破らない限り実行できない

拡張されたChurch-Turing Thesis

□ The extended CT thesis:

Turingマシンで、多項式時間で効率的に実行できない計算は、いかなる物理システムによっても実行できない。

あるいは同じことだが、

物理システムによって実行されるすべての計算は、あるTuringマシンによって多項式時間で実行される。

という命題「拡張されたChurch-Turing Thesis」を考えよう。

□ しかし、Turingマシン(古典コンピュータ)で、多項式時間で効率的に実行できない計算(例えば素因数分解)が、量子コンピュータで多項式時間で解けるなら(Shorのアルゴリズム)、この命題は成り立たない。

□ 「量子超越性」は、拡張されたChurch-Turing Thesisが成り立たないことを示す。

Church-Turing Thesisの量子化

- 次のような、Church-Turing Thesisの量子版を考える。

The Quantum-extended CT thesis (qECT):

量子Turingマシン(あるいは量子回路)によって実効的に実行できない全ての計算は、物理法則と矛盾しない、いかなる物理システムによっても実効的に実行されない。

すなわち、

物理システムで実効的に計算可能な計算は、量子Turingマシン(あるいは量子回路)によって実効的に実行される。

- これは、正しそうだ。
- ただ、Susskindによると、ブラックホールに落ち込んだ時、これも成り立たなくなり、修正が必要となるという。

Church-Turing Thesisの量子化の修正版

- **The Modified Quantum-extended CT thesis:**
量子Turingマシン(あるいは量子回路)によって実効的に実行できない全ての計算は、空間のホログラフィック境界とコミュニケーション可能なままにある、いかなる物理システムによっても実効的に実行されない。

Church-Turing Thesis をめぐる 最新の議論

- Church-Turing Thesis を巡るこうした議論については、次のSusskindの論文を参照されたい。
- ここでは、Church-Turing Thesis を巡る議論が、数学的な計算理論の枠を超えて、物理学の量子重力理論との関係で論じられている。

“Horizons Protect Church-Turing”

Susskind 2020年5月

<https://arxiv.org/pdf/2003.01807.pdf>



“COMPLEXITY” のロゴの入ったTシャツを着る Susskind

現代物理学の主要な関心領域

量子多体系の物理
(場の理論、物性理論、統計力学、可解系)

AdS/CFT
(ホログラフィー)



重力理論
(超弦理論)

量子エンタングルメント
テンソルネットワーク

量子情報理論
(情報幾何)



BH情報問題

高柳匡 「重力理論と量子エンタングルメント」
<https://goo.gl/kRPcNI>

$$\text{MIP}^* = \text{RE}$$

“MIP* = RE”

Zhengfeng Ji, Anand Natarajan, Thomas Vidick,
John Wright, Henry Yuen

2020年1月

<https://arxiv.org/pdf/2001.04383.pdf>



Henry Yuen



Thomas Vidick



Zhengfeng Ji



Anand Natarajan



John Wright

論文 “MIP* = RE”

□ **MIP* = RE :**

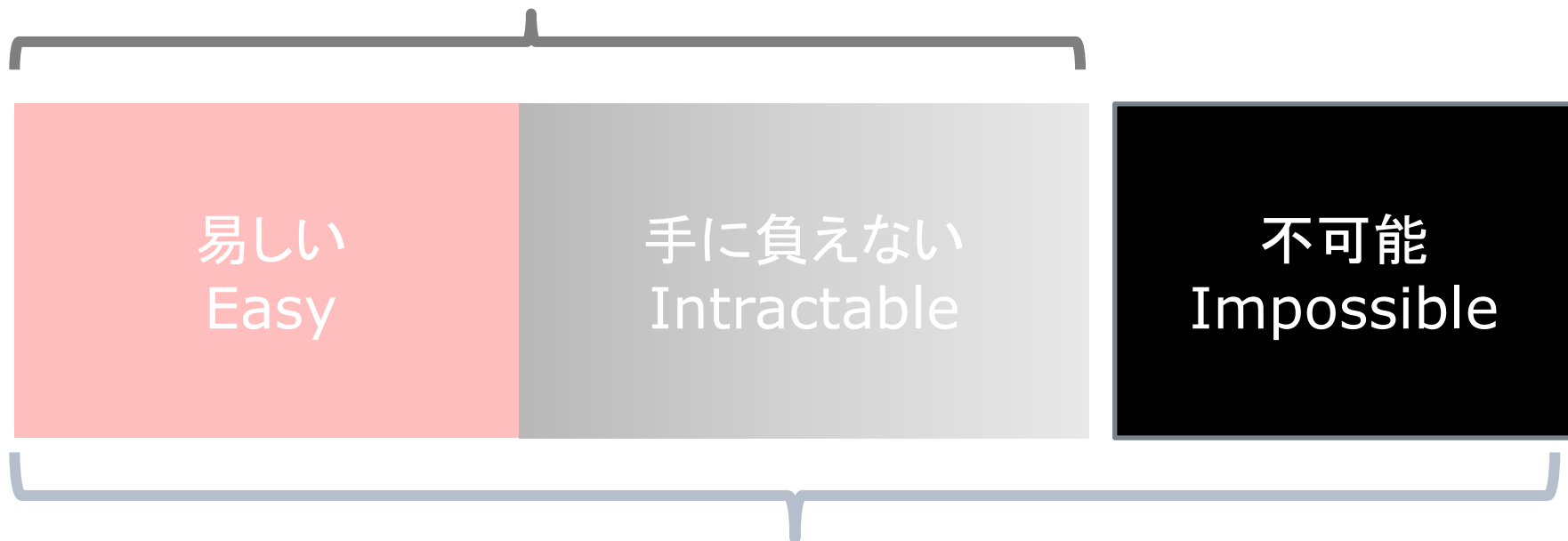
We show that the class MIP* of languages that can be decided by a classical verifier interacting with multiple all-powerful quantum provers sharing entanglement is equal to the class RE of recursively enumerable languages.

□ **Halting Problem :**

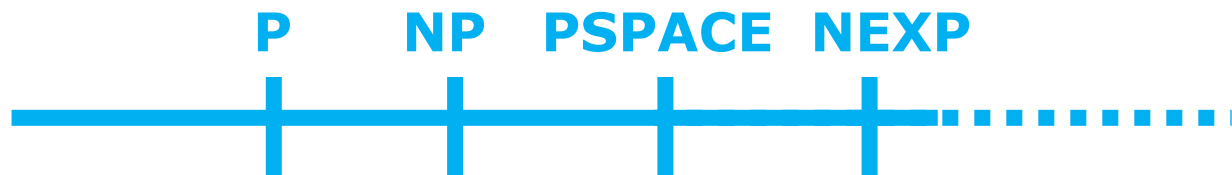
An immediate byproduct of our result is that there is an efficient reduction from the Halting Problem to the problem of deciding whether a two-player nonlocal game has entangled value 1 or at most $1/2$.

MIP*=RE 以前

計算複雑性理論



計算可能性理論



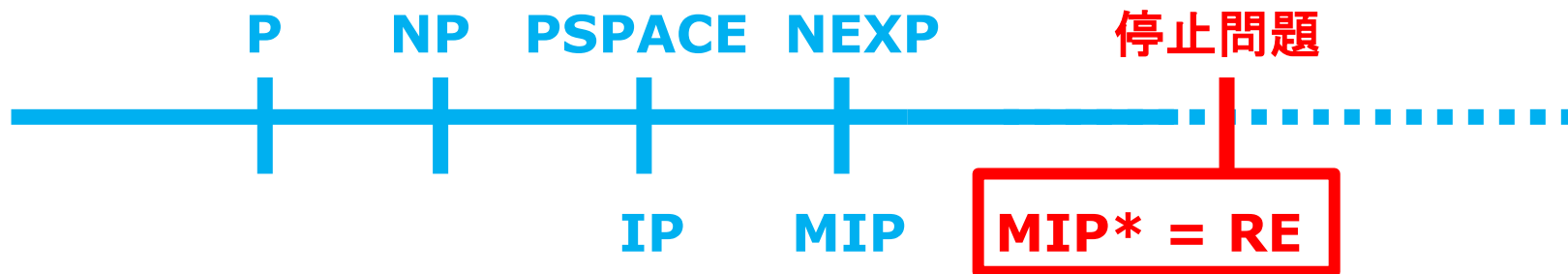
複雑性のクラス

MIP* = RE 以降の

計算複雑性理論



計算可能性理論



論文 “MIP* = RE”

□ **Tsirelson's problem :**

Using a known connection, undecidability of the entangled value implies a negative answer to Tsirelson's problem: we show, by providing an explicit example, that the closure C_{qa} of the set of quantum tensor product correlations is strictly included in the set C_{qc} of quantum commuting correlations

□ **Connes' embedding conjecture :**

Following work of (Fritz, Rev. Math. Phys. 2012) and (Junge et al., J. Math. Phys. 2011) our results provide a refutation of Connes' embedding conjecture from the theory of von Neumann algebras..

計算科学

量子力学

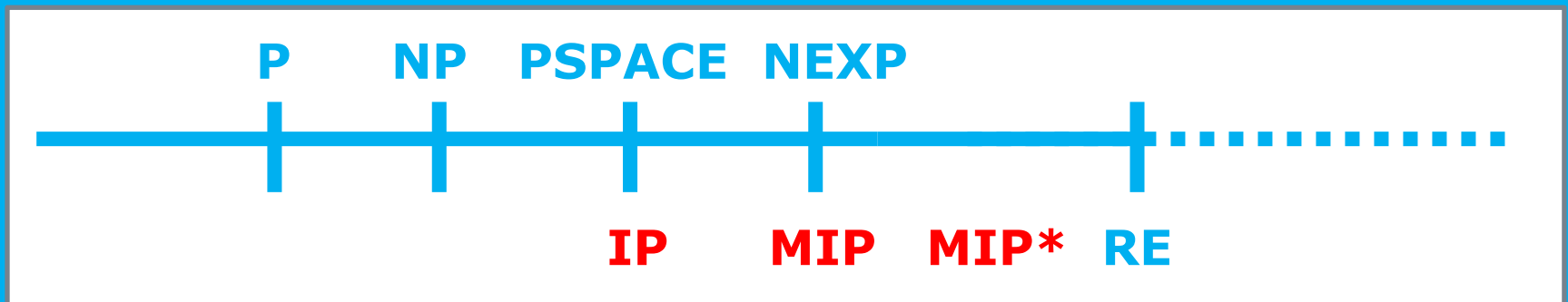
純粋数学



MIP* = RE定理の射程

Interactive Proof

IP, MIP, MIP* という複雑性のクラスの名前に含まれる "IP" はInteractive Proofを表す。



対話でわかること

M

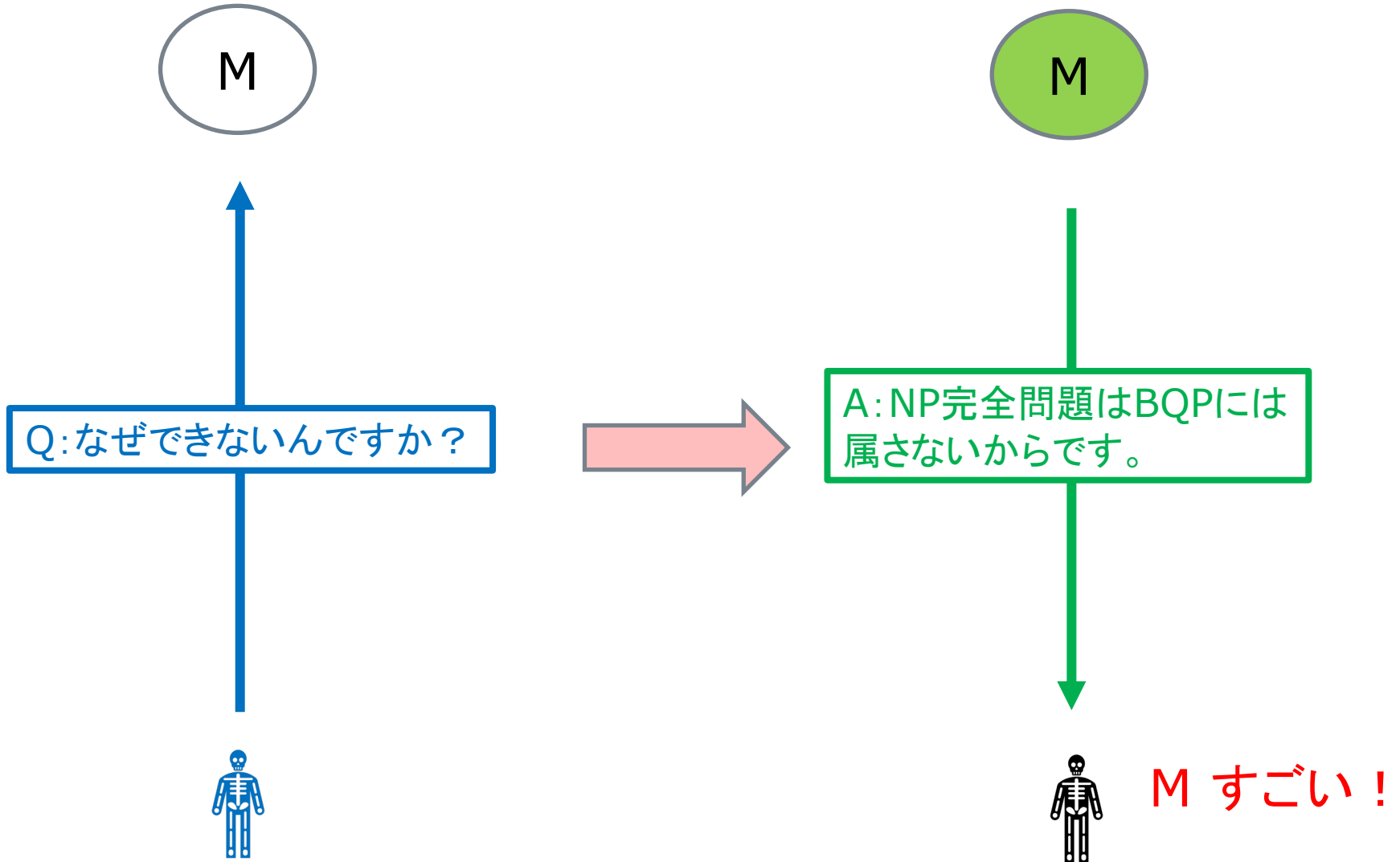
M

Q: 量子コンピューターを使用して、NP完全問題を多項式時間で解決することは可能ですか？

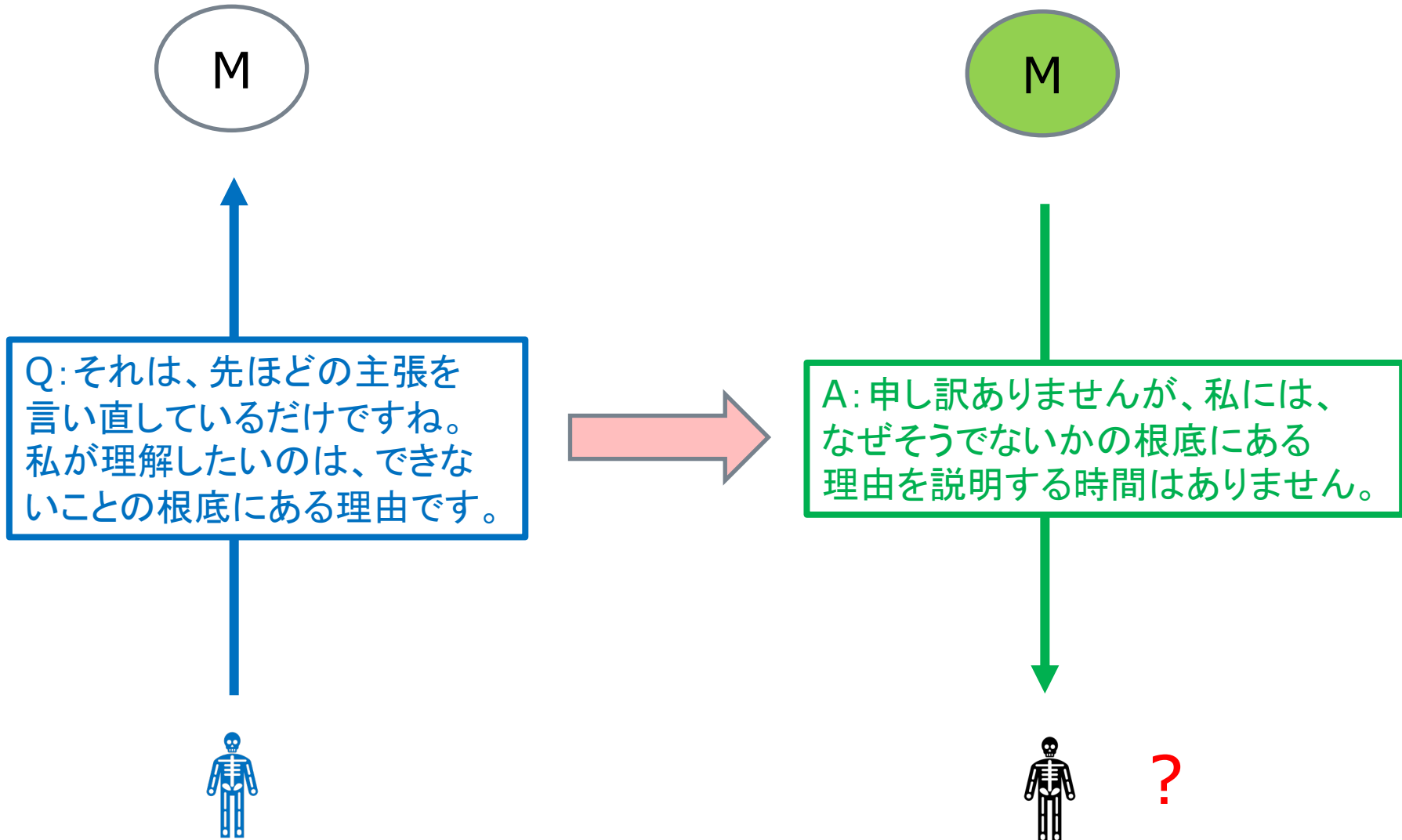
A: 量子コンピューターを使用してNP完全問題を多項式時間で解くことはできません。



対話でわかること



対話でわかること



対話でわかること

M

Q: 怒ってるわけじゃないけど、それは本当に嘘くさいですね。あなたはその理由を分かっていると思います。



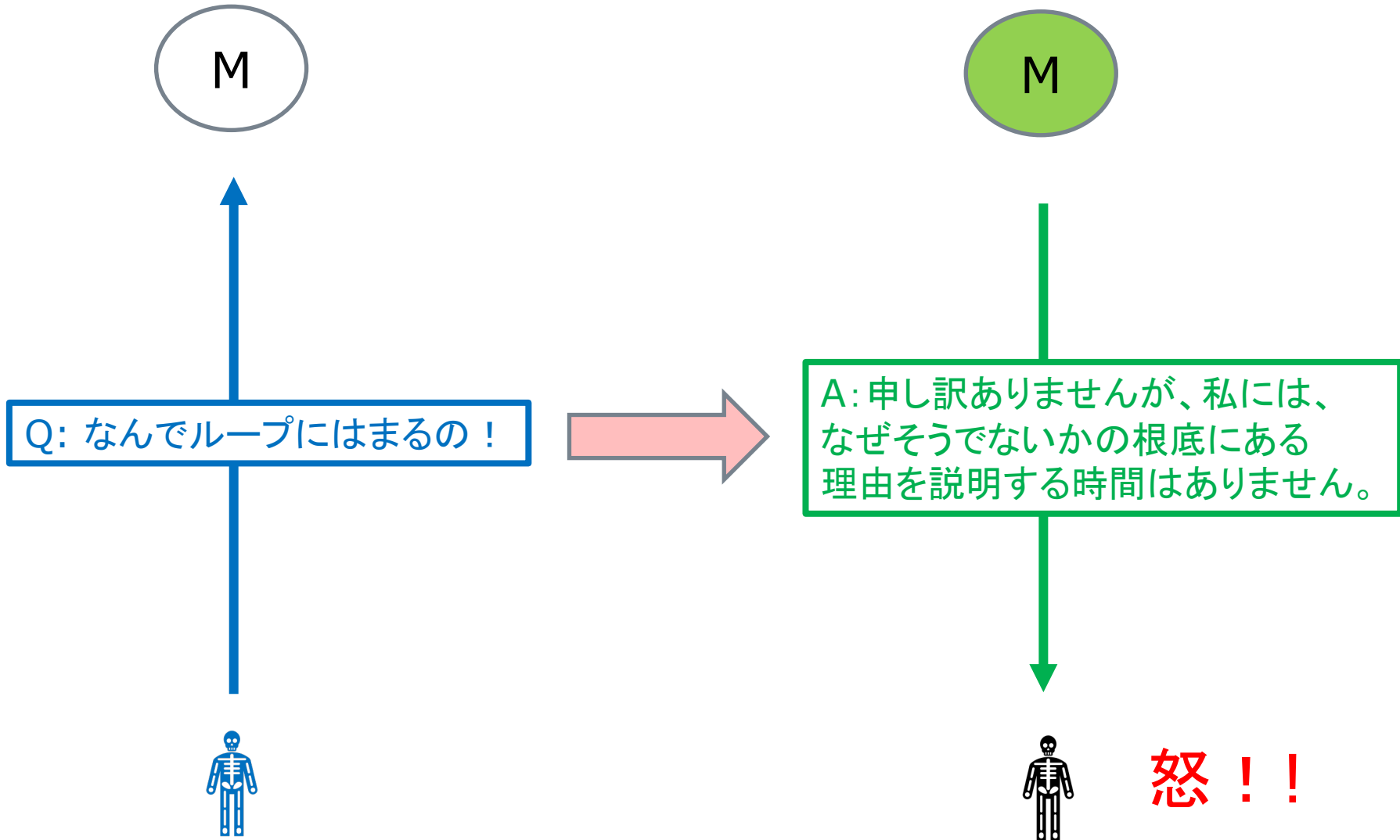
M

A: 申し訳ありませんが、私には、なぜそうでないかの根底にある理由を説明する時間はありません。

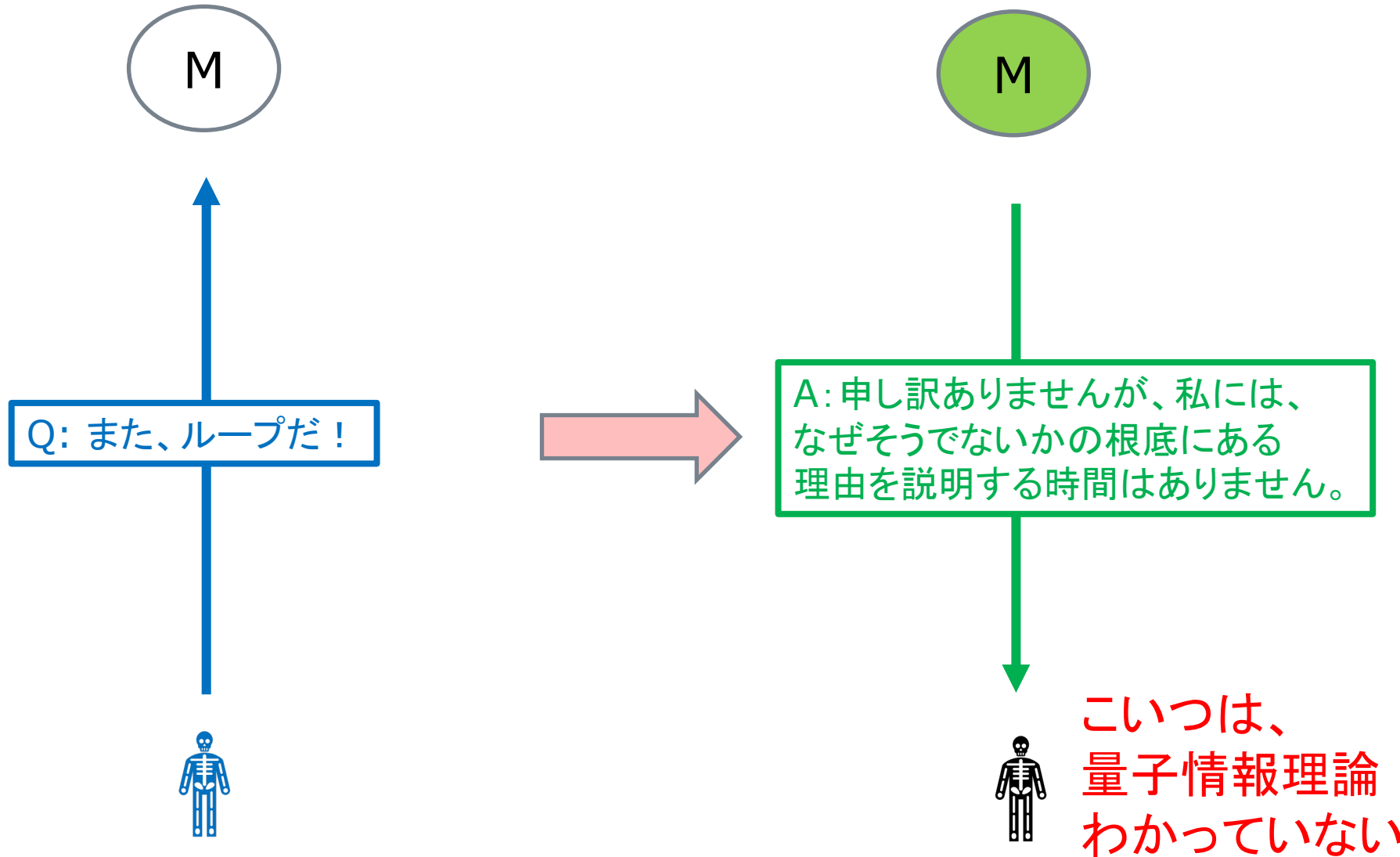


怒！

対話でわかること



対話でわかること



ArthurとMerlin

- AとBは、同じ大きさのボールである。Aは赤い色でBは緑色をしている。ただ、両者の重さは違っている。Aは重く、Bは軽い。
- Arthurは目が見えない。二つのボールが、どんな色をしているかは分からない。ただ、手に持って重さで二つのボールを区別できる。
- 明らかに、Arthur一人ではA,Bの色を知ることはできない。ここで、もう一人 Merlinの助けを借りることにする。
- Merlinは、ボールには触れずに、Arthurの質問だけに答えることとする。ただし、Merlinは、嘘をつく。もっと悪いことに、Merlinは盲人かもしれないし、色盲かもしれない。
- この状況で、Arthurは、二つのボールの色を知ることができるだろうか？

ArthurとMerlin

- Merlinは嘘つきだが、ということ全てが嘘という訳ではない。MerlinはArthurの質問に対して、知っていることについては、3回に1回程度の嘘しかつかない。3回に2回程度は正しいことを言う。知らないことについては、適当な答えを返すとする。
- 嘘つきMerlinの嘘についての上の条件だけで、実は、盲人Arthurは二つのボールの色を知る可能性があるのである。
- 最初にArthurがチェックすべきことは、Merlinがボールの色を認識できているかのチェックである。Merlinが色の認識ができないのなら、もちろん、Arthurは、Merlinの助けでは、ボールの色を知ることはできない。
- ただ、Merlinが色の認識ができていることがわかれば、Arthurはボールの色がわかる。
- **どうやって？ 各自考えること。**

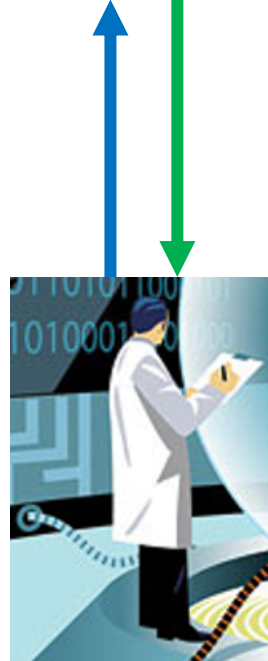
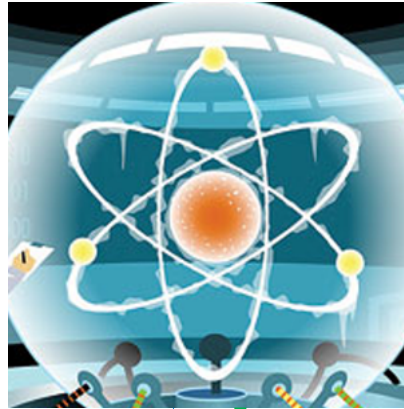


嘘つきMerlin

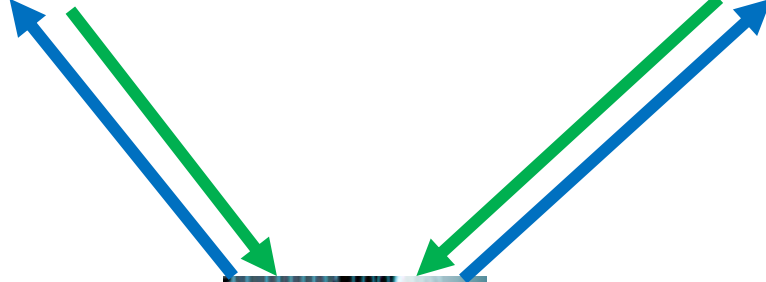
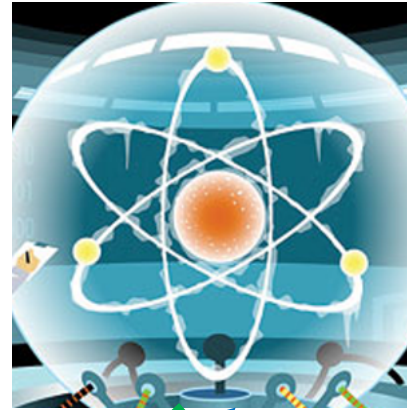
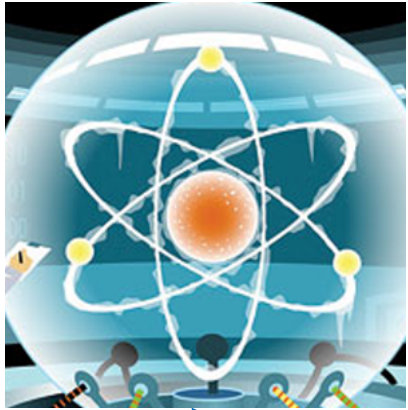
人工知能論と Interactive Proof

IP, MIP, MIP* という
Interactive Proofで利用される
「対話」のモデルをまず見てみよう。
人工知能論としてその意味を考える。

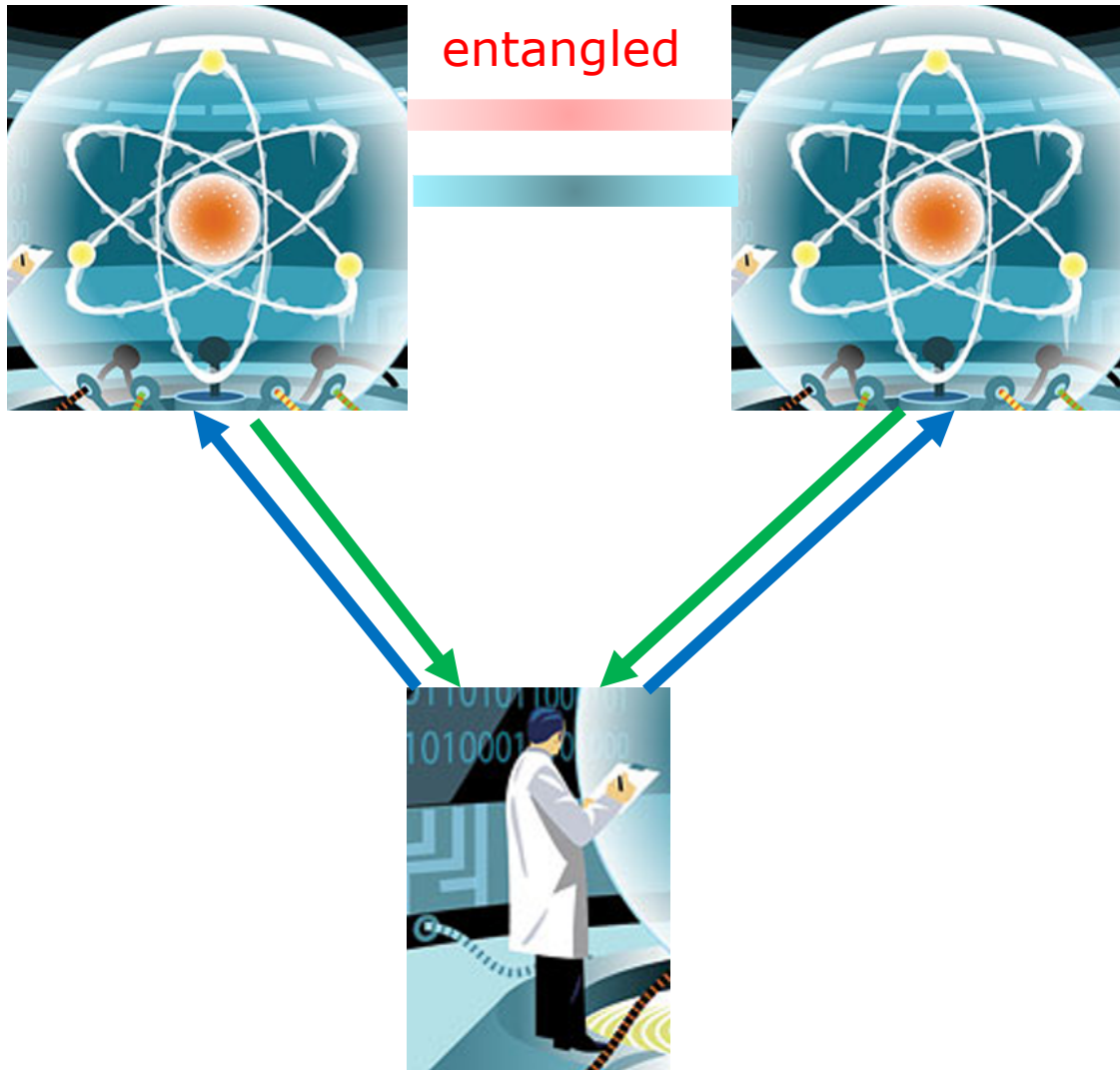
IPのモデル



MIPのモデル



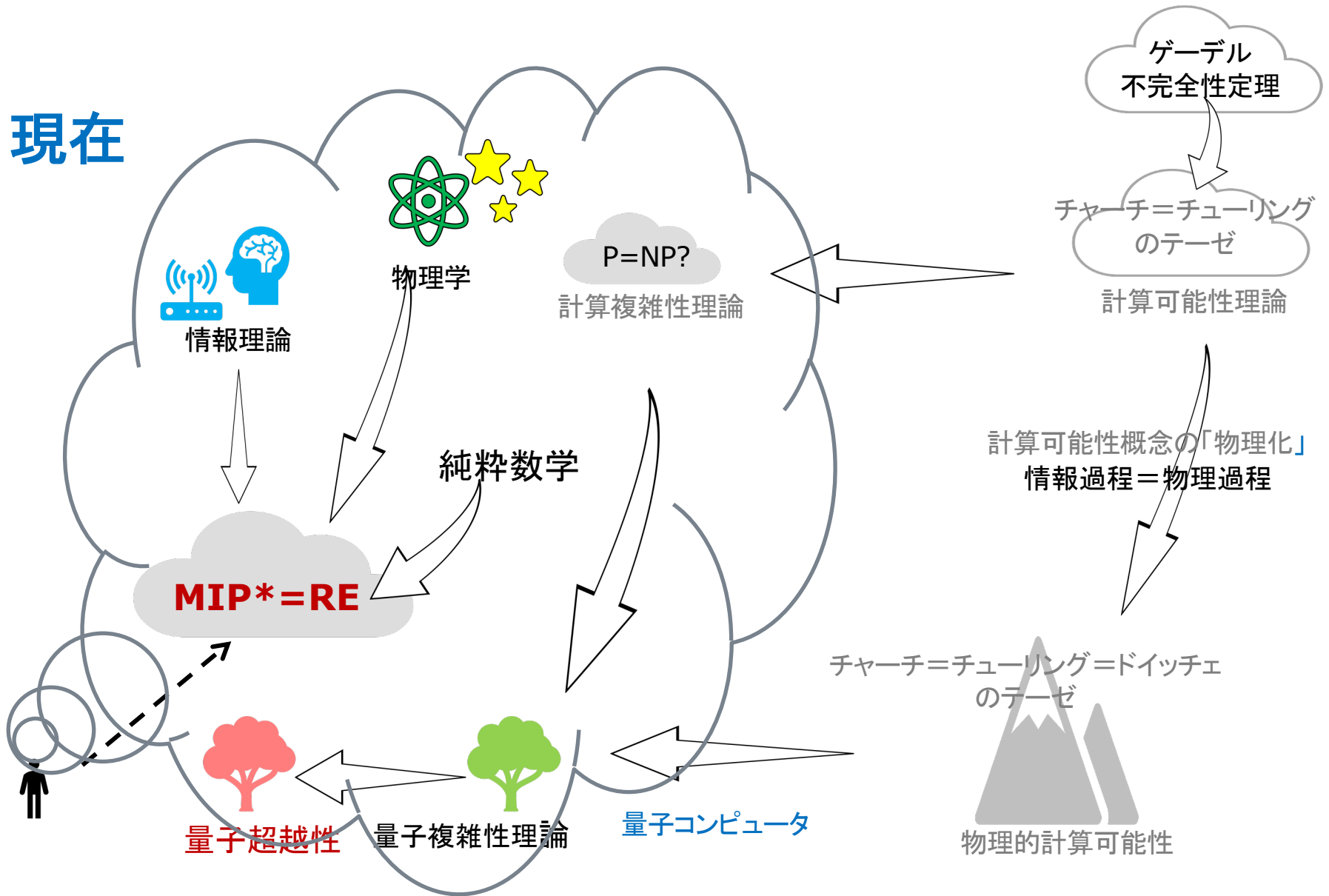
MIP*のモデル



人間と機械の両者が構成する系 として人工知能の可能性を考える

- 先に見たように、量子コンピュータが、計算のモデルとしての Turingマシンを上回る計算能力を持つのなら、機械としての量子マシンは、人間の知能を超える能力を持つことになる。
- ただし、人間が単独では Turingマシンと同程度の認識能力しか持たないとしても、人間は人間以上の能力を持つ機械との対話を通じて、人間の認識能力を拡大できる。
- それは盲人の Arthur が、Merlin の助けを借りて色の認識ができるようになることと同じである。
- MIP* のモデルは、人間と機械の両者が構成する系として人工知能の可能性を考えるフレームを提供するものとして解釈できる。

現在



参考資料

- 数学的認識について / 数学の基礎と計算科学
<https://www.marulabo.net/docs/math-foundations/>
- $MIP^* = RE$ 関連資料集
<https://www.marulabo.net/docs/mip-re/>
- 9/30 マルゼミ
「人間と機械の共生」で理解する Interactive Proof