

大規模な計算資源の 強いゲームプレイへの貢献

美添 一樹

九州大学 情報基盤研究開発センター 教授

2022年4月8日 α xSCシンポジウム「ゲームとスーパーコンピュータ」

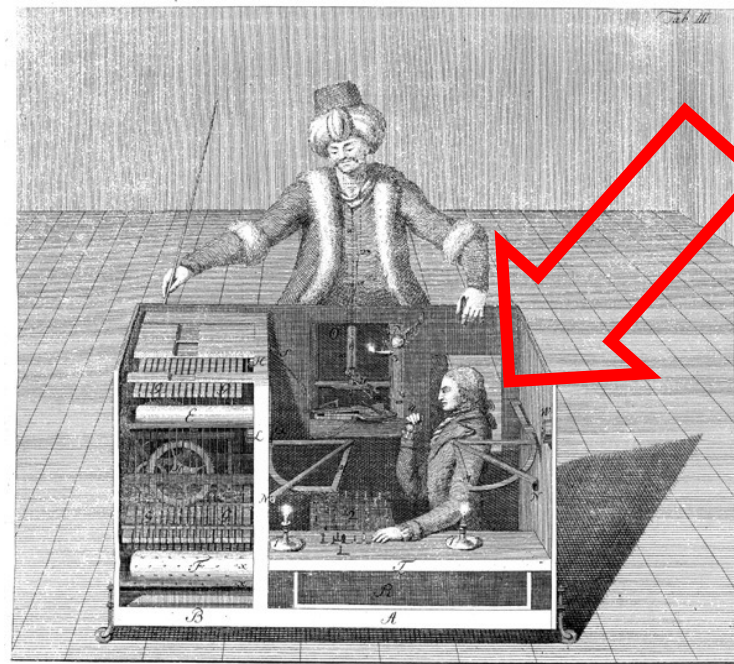
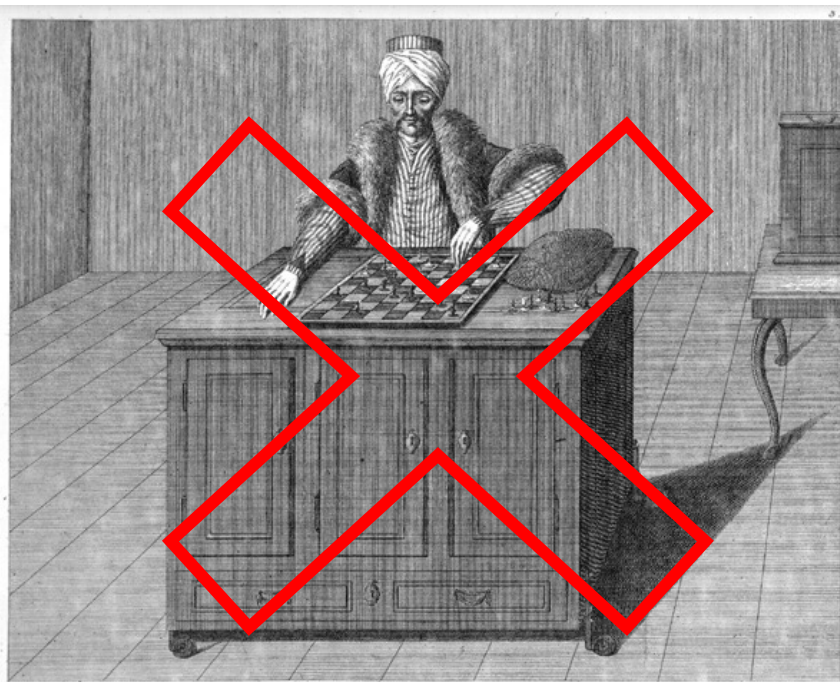
<https://www.cc.kyushu-u.ac.jp/scp/users/news/2021/523.html>

頭の良い機械を作りたい

という夢は昔からあった

The Turk (トルコ人) 1770年~1854年

チェスを指すオートマタ(機械人形)



W. de Kempelen, del. *Cher à Meckel, exco.*, Basilea. P. G. Pinz, sc.
Der Schachspieler im Spiels-begriffen. Le Joueur d'Échecs tel qu'on le voit pendant le jeu.

コンピュータ
DeepBlue (IBM)

対 チェス世界チャンピオン
Kasparov (ロシア)



1997年
チェス対戦
DeepBlue勝利

ゲーム木探索

チェス専用チップ
利用のスパコン

(今から見ると力任せ)



Copyright 2007, S.M.S.I., Inc.
Owen Williams, The Kasparov Agency.

AlphaGo vs プロ囲碁棋士

AlphaGo

Google Deep Mind 社
が開発した囲碁プログラム

2016年3月

イ・セドル（元世界最強）に4勝1敗で勝利

2017年1月

ネット上でトップ棋士に60連勝
(ハンドル名 Master)

2017年5月

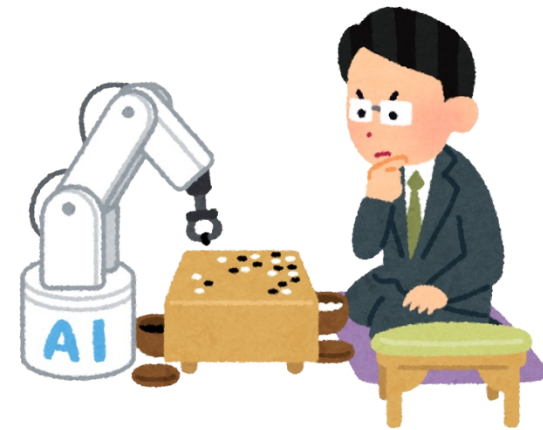
柯潔（現世界最強）に3連勝

2017年10月

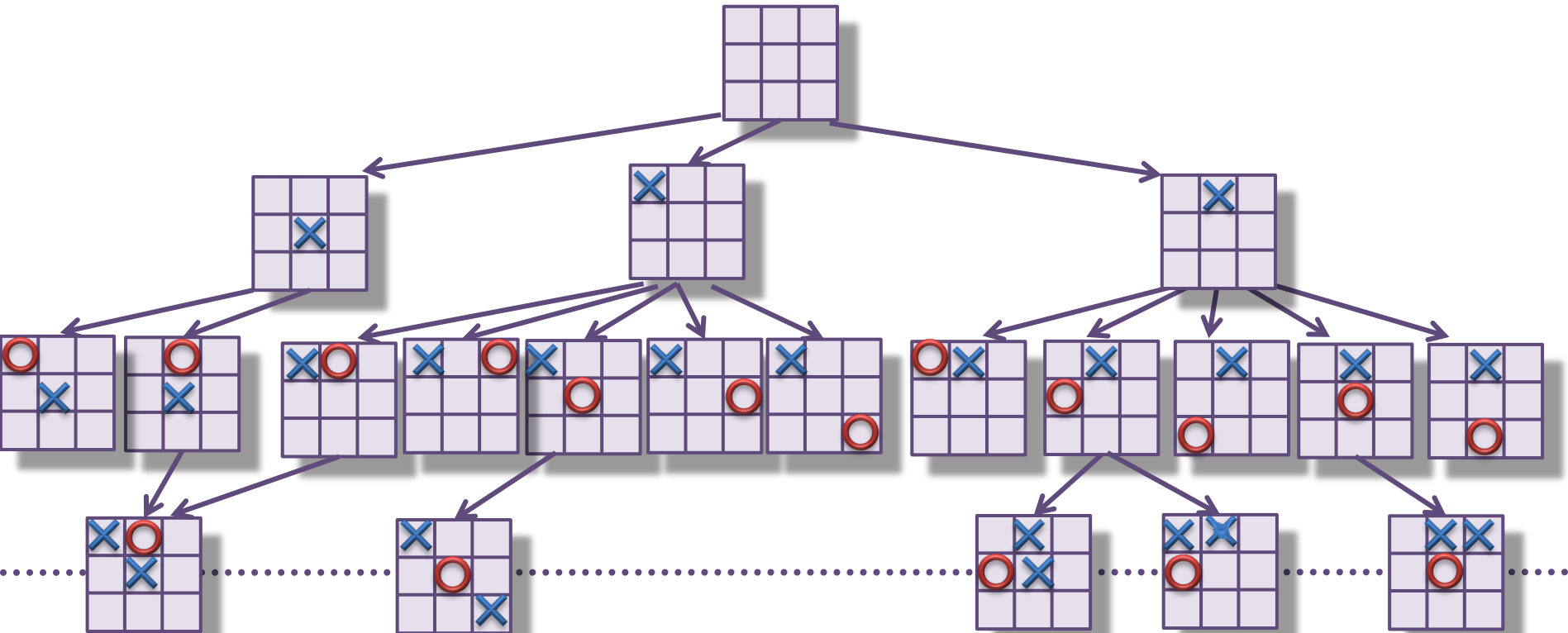
囲碁のルールだけから学習した
AlphaGo Zero が AlphaGo を上回る

2017年11月

AlphaZero 同手法で将棋とチェスも最強に



念のため: ゲーム木探索



ゲーム木探索の2つの方法

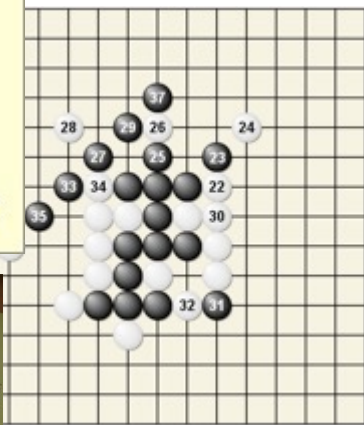
その1, 最後まで読み切る

その2, 評価関数を使う

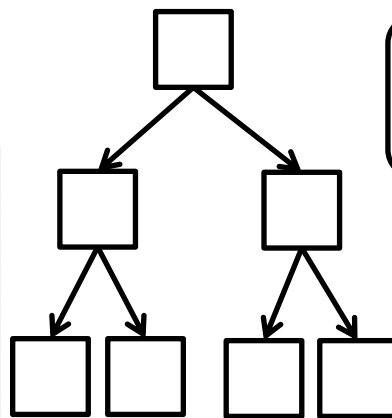
伊藤香寿 寿 811手

詰将棋

五目並べ



チェッカー



? ? ? ?
どの手が
良いのか?

局面



スコア

がんばって
良い評価関数を作る

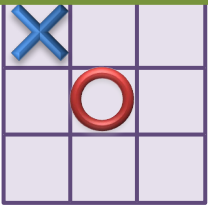


どうぶつ
しょうぎ

その1, 最後まで読み切る

読み切れるゲーム、無理なゲーム

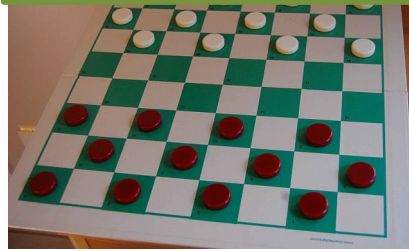
三目並べ
765



どうぶつしょうぎ
246,803,167



チェッカー
 5×10^{20}



ゲームの
局面数

読み切れる



オセロ
 10^{28}



チェス
 10^{41}



将棋
 10^{69}



囲碁
 10^{170}



読み切れない

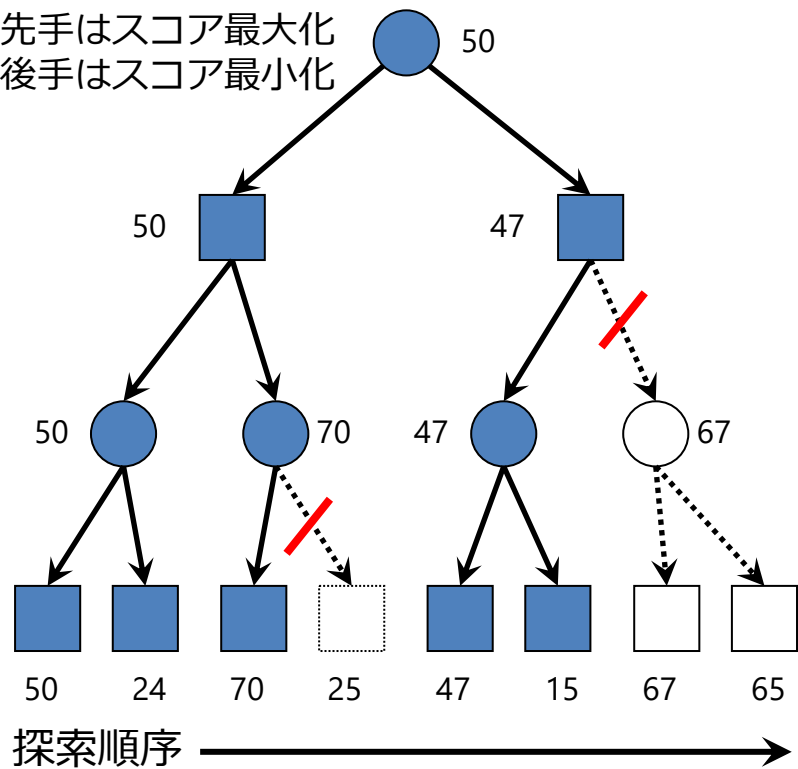
注意: これは合法局面の数 (の推定)
手順を考慮するとさらに大きい (主に千日手の影響)

つまり評価関数が必要

$\alpha\beta$ 探索

(minimax探索 + $\alpha\beta$ 枝刈)

先手はスコア最大化
後手はスコア最小化



探索を途中で打ち切り
評価関数の値を用いる

minimax探索 + 手作り評価関数の成功例は多数存在する

熟練の開発者の手によるいわば「芸術品」



チェッカー

1994年にチャンピオンに勝利
CHINOOK vs Tinsley

オセロ

1997年にチャンピオンに勝利
Logistello vs 村上



チェス

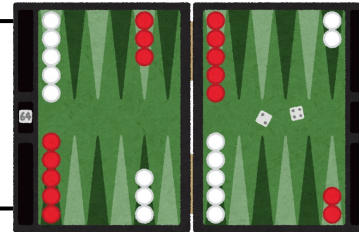
1997年にチャンピオンに勝利
IBM DeepBlue vs Kasparov

機械学習の時代

機械学習による評価関数の成功例（引用は初出の論文ではない）

[Tesauro, Gerald 1995] "*Temporal Difference Learning and TD-Gammon*"

TD-gammon
1992年 強化学習+NNによる
バックギャモンプログラム



[Buro 1995] "*Logistello: A Strong Learning Othello Program*"

オセロ
1997年にチャンピオンに勝利
Logistello vs 村上



[Hoki, Kaneko 2014] "*Large-Scale Optimization* for Evaluation Functions with Minimax Search"

将棋
2005年 Bonanza method
現在は人間よりかなり強い



学習による評価関数

棋譜データからの
機械学習による評価関数

主に2つ方法がありうる

- 1, 勝敗 / 形勢を推測する
- 2, 最善手を予測する

ここではまだ深層学習の
話題ではなく、
より単純なモデル

基本的なアイデア

ある局面 p の
特徴ベクトル

$f(p)$

それぞれの
特徴の重み

w

ある局面 p の評価 $e(p, w)$

$$e(p, w) = \sum_i w_i f_i(p)$$

将棋における「特徴」と「重み」の例

飛車がある (特徴) x 飛車の価値 (重み)

玉と金銀が特定の配置にある x その重み

重みの次元と学習手順

駒の価値	13通り
玉と駒一つの位置関係	81 x 駒の種類 x 80 合法手と対称性より 60,876
玉と王と駒一つの位置関係	81 x 80 x 駒の種類 x 79 合法手と対称性より 2,425,950
玉とそれ以外の駒二つの位置関係	81 x 駒の種類 x 80 x 駒の種類 x 79 合法手と対称性より 44,222,454

これはあくまで一例でもっと大量の特徴量を用いる例も存在する

$$e(p, \mathbf{w}) = \sum_i w_i f_i(p)$$

f, \mathbf{w} の次元が数百万～数億

訓練例の全局面につき、
浅い探索を行って候補手を選ぶ

プロの着手との不一致を数える

目的関数の偏微分に基づいて
重みを更新する

収束するまで
右の手順を
繰り返す

詳細は省略

最初のbonanzaの学習は
Pentium 4 のマシンで90日

計算時間が強さに変換される時代

新たな枠組みとして
機械学習による評価関数 + $\alpha\beta$ 探索
「工芸品」から「工業生産品」へ

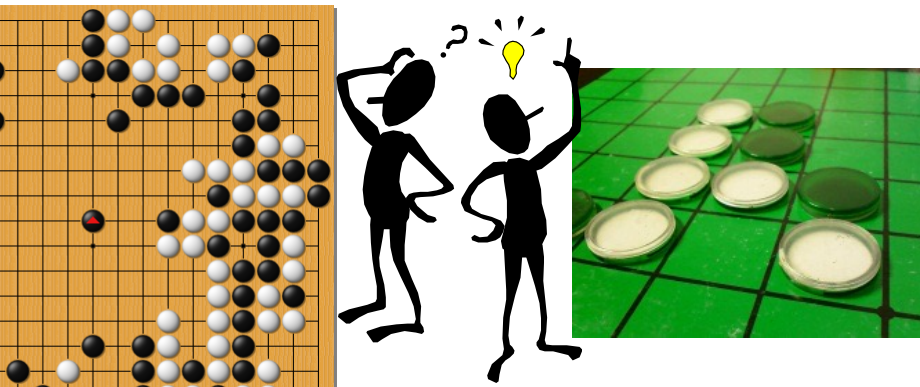
囲碁の評価関数は作れるか？



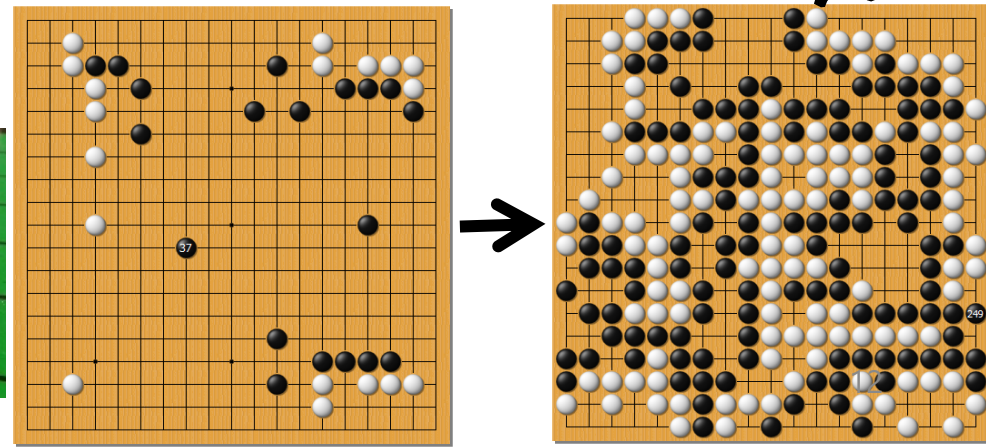
良い評価関数は、
正確で高速でないといけない

深層学習以前は
囲碁の**評価関数**を
作れた人は居なかった

オセロの隅ほど
特徴のある場所がない



途中で終局図を
予想するのも困難

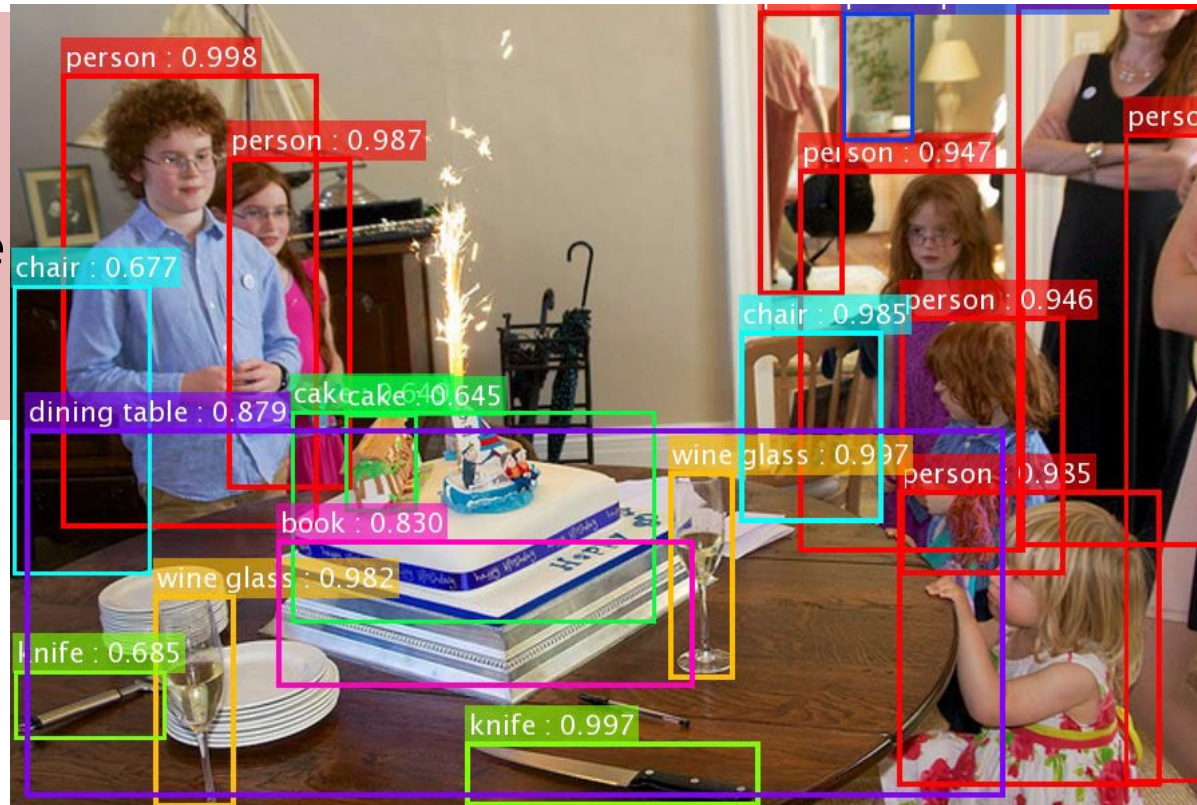


コンピュータの画像認識

ILSVRC

Imagenet large Scale
Visual Recognition Challenge
プログラムによる
画像認識の精度を競う

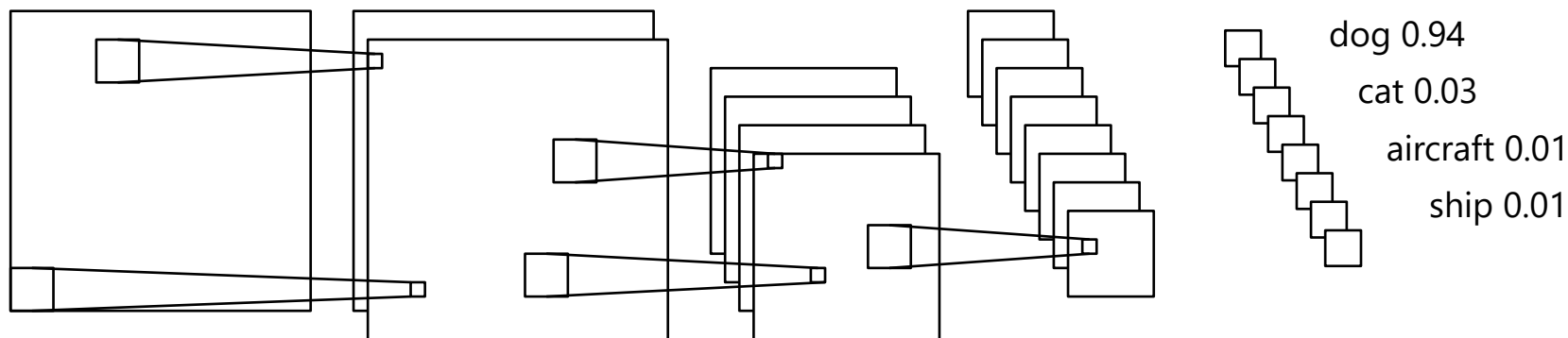
犬と猫を見分ける
くらいは余裕



[K. He et al. 2015, Microsoft Research Asia]

2014年 人間（非専門家）を超える精度を達成
（Google と Microsoft）

DCNN: Deep Convolutional Neural Network



フィルタをたくさん重ねると、複雑な形状を検知できる
個々のフィルタは他純な操作に対応（輪郭線抽出とかぼかしとか）

画像認識では例えばこの数値が高いと猫、
と言うのが特定されている (by google)

<https://googleblog.blogspot.jp/2012/06/using-large-scale-brain-simulations-for.html>

Microsoft の ResNet は 152層 [K. He et al. 2016]
(注：1,202層まで実験されているが、最高精度は152層)



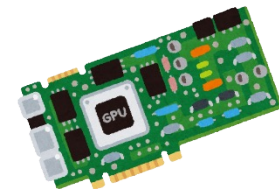
「猫顔ニューロン」

画像認識の手法で着手予測

囲碁は19x19の画像

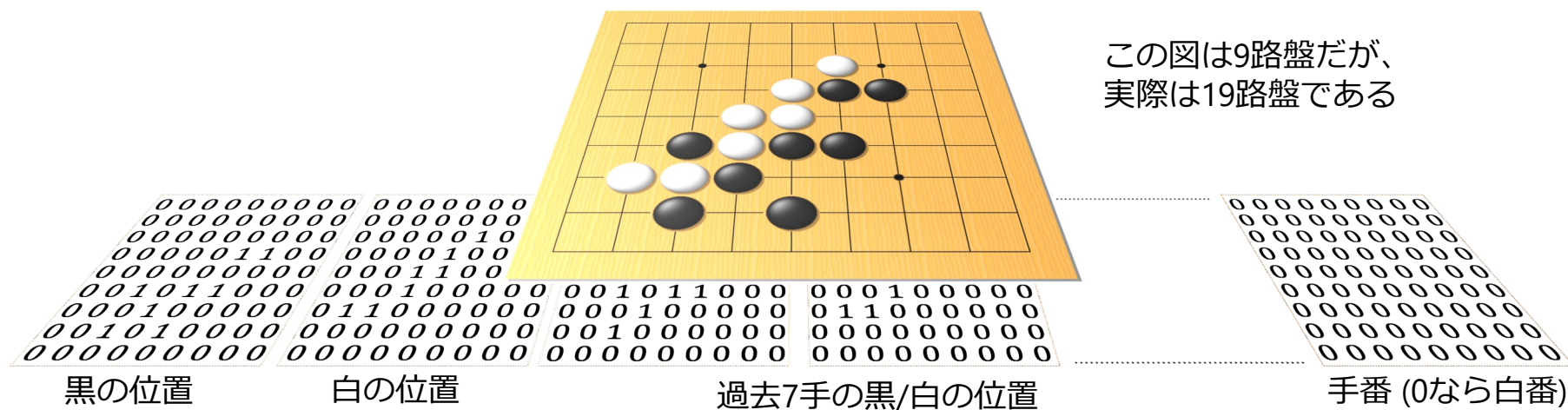
盤面から画像を生成し、「犬」「猫」などの代わりに次に打つ場所を予測
(通常の画像はRGBの3枚、AlphaGo Zero では下図の17枚)

アマ高段者の棋譜
約17万局から学習



予測精度57-60%

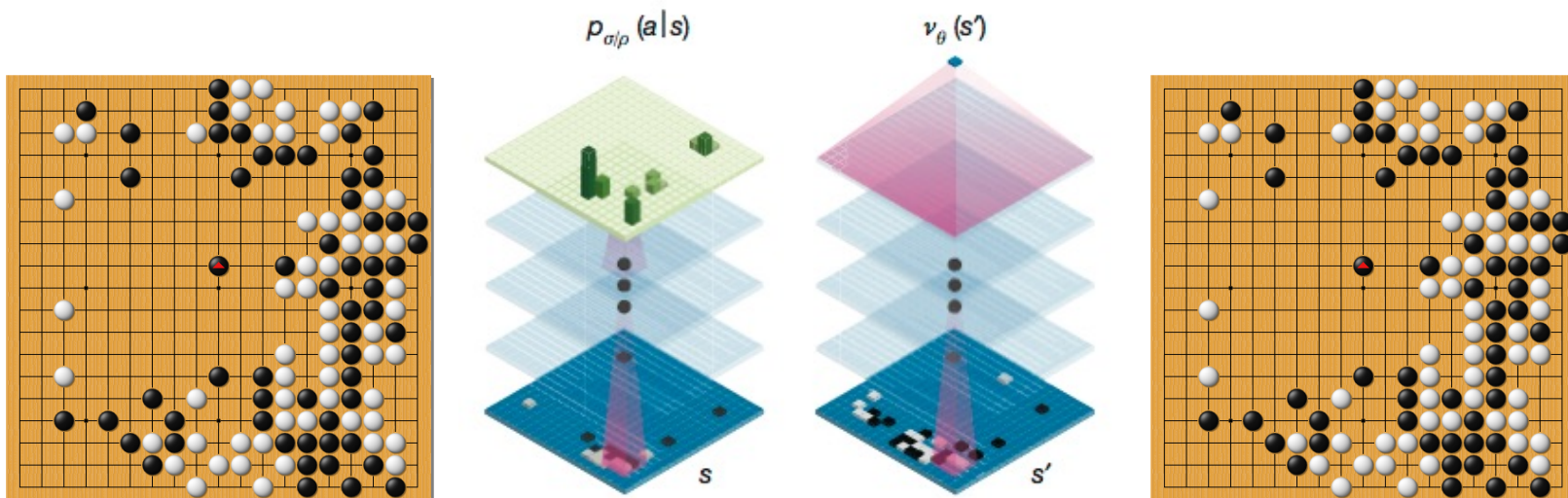
アマ5段が試したら50%程度



policy network

value network

「着手予測」と「形勢評価」



[Silver, Huang et al. 2016] Fig. 1b

policy network は
局面の着手確率を予測

value network は
局面の勝率を予測

人間の棋譜からの
教師あり学習で実現可能
最新のGPUなら1枚で数週間？

1つの対局の約200手全てが教師例
(棋譜数 x 約200 の教師例)

着手予測よりも大量のデータが必要

同じ対局の局面は類似している
そのまま使うと過学習する
(棋譜数 x 1 の教師例)
(実際は1はちょっと極端だが)

自己対戦で棋譜生成→学習

Deep Learning の問題点

データが大量に必要

理想的な解決策

プロ棋士を1万人雇用して
ひたすら対局させる（無理）

AlphaGo の解決策

まずプロ棋士の手を予測するDNNを用意

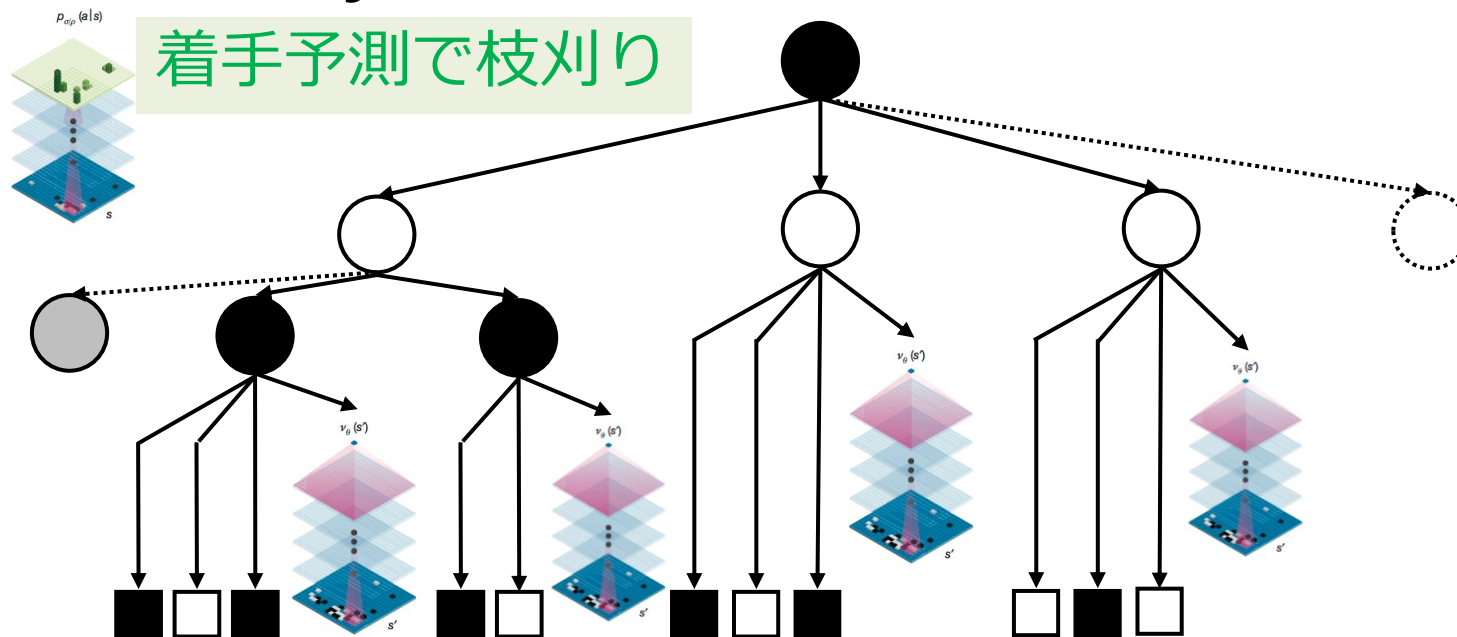
それを強化学習でちょっと強くする（アマ5段程度）

それを使って**3,000万局**対局させ、**1局から1局面**採取

それを元に Deep Learning で **Value Network** 作成

AlphaGo 初期バージョン

policy-net + value-net + MCTS



Monte-Carlo Tree Search を用い、
value-net の評価 (予測勝率) の高い方へ探索木を伸ばす
なお、初期バージョンはNN以外にサンプリング (rollout) も併用

このバージョンが、元世界ランク1位の囲碁棋士
イ・セドルに4勝1敗で勝利 (2016年3月)

AlphaGo Zero

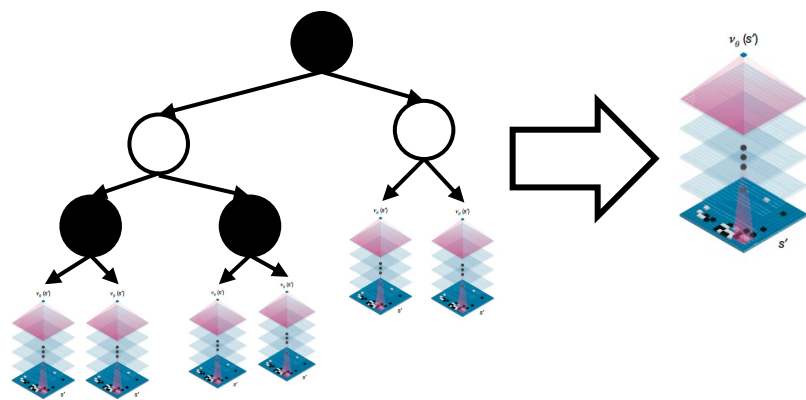
人間の棋譜を全く使わない
"Zero" からの学習

自己対戦による強化学習
膨大な計算資源

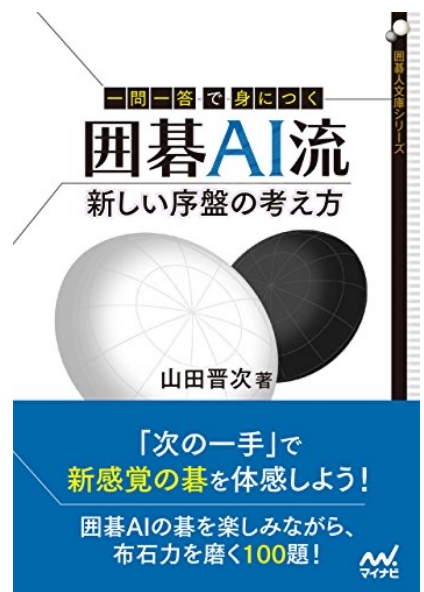
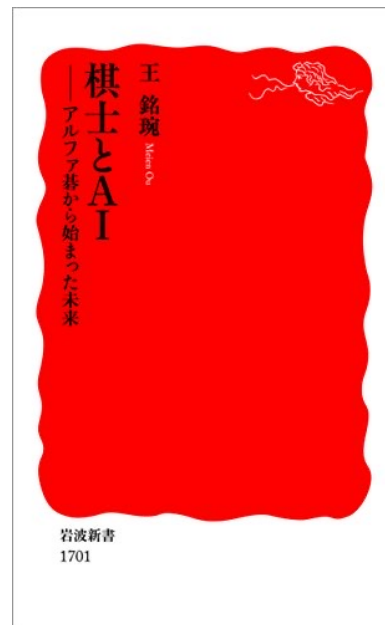
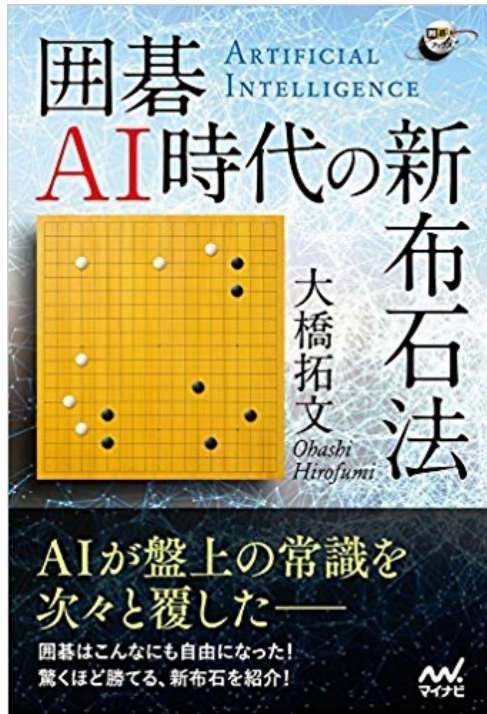
深い探索で自己対戦した結果を
予測するようにモデルを訓練
(基本のアイデアは単純)

しかし自己対戦の規模が巨大
学習の各ステップで
25,000局の自己対戦 (1手0.4秒)
(1GPUなら20日ぐらい)

最終的に40日間で2千9百万回
の自己対戦を行う
(1GPUなら 25,000日ぐらい)



前のバージョンの AlphaGo
に対して 100戦全勝
人間に負けた記録は無し



探索も並列化の対象

「早さは強さ」。以下はプロに勝利した例

チェス専用チップ搭載
スーパーコンピュータ
(今から見ると力任せ)

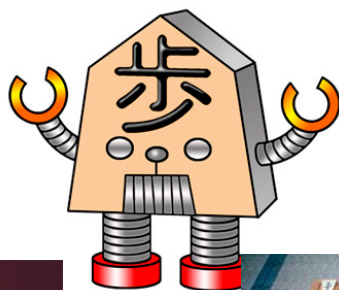
IBM Deep Blue
[対Kasparov 1997]



普通の計算機・
普通のネットワーク
だが効率よい並列化
(プログラミングの
芸術だと思えます)

GPS将棋

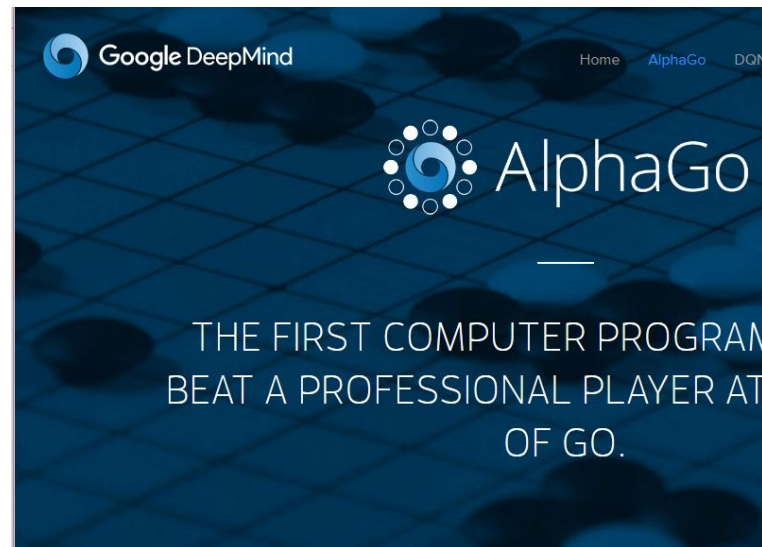
[第2回電王戦 2012]



Monte-Carlo Tree Search
を並列化
176GPUを使用
(対1ノード版の勝率 75%)

AlphaGo

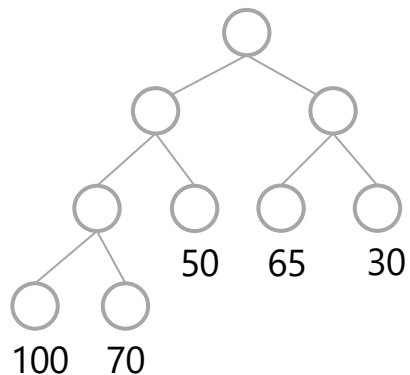
[対イ・セドル 2016]



Monte-Carlo Tree Search (MCTS)

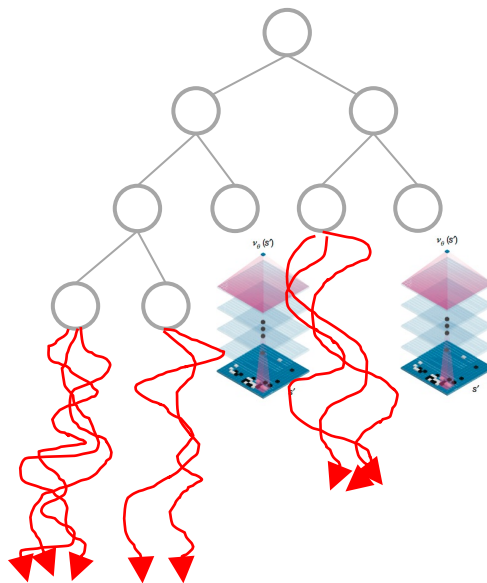
確率的な探索アルゴリズム。強化学習と共通の理論的背景を持つ

通常の探索アルゴリズム



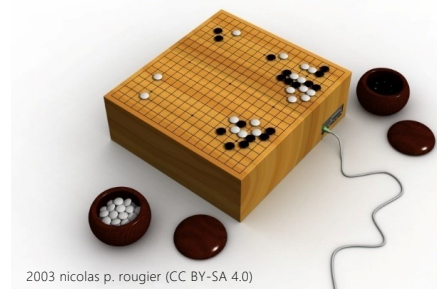
節点の評価値はスカラー
大小比較で探索

Monte-Carlo Tree Search



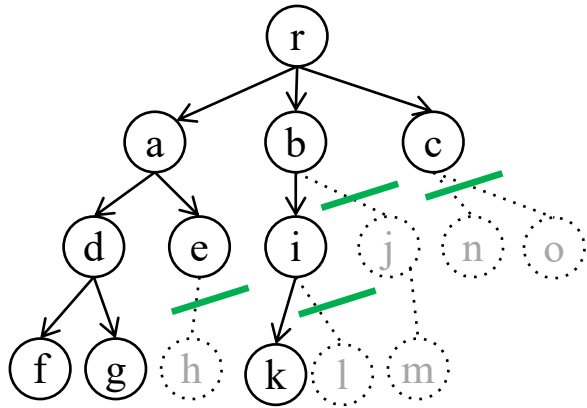
Upper Confidence Bound [Auer+2002]
MCTS [Coulom2006]

節点の評価値は“確率”
サンプリングや機械学習で
確率を得る
確率同士を比較して探索



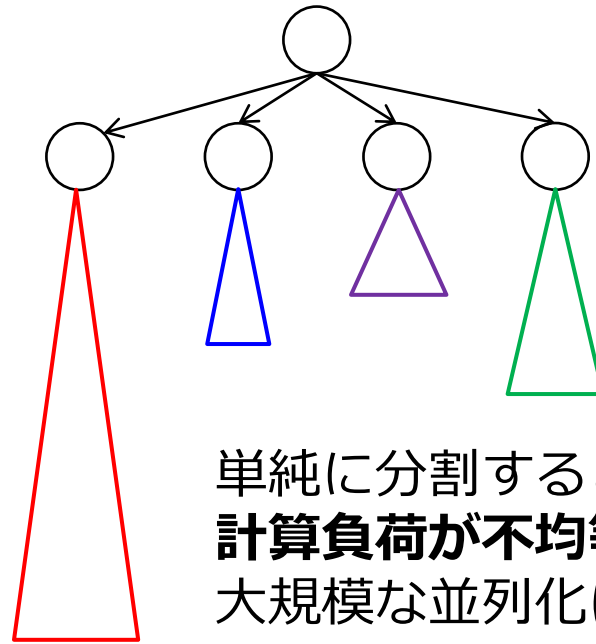
元々、囲碁の研究から生まれた手法
ゲーム以外にも非常に多彩な応用あり
AlphaGo も利用している

並列探索の難しさ

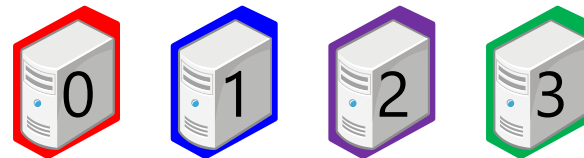


探索空間が巨大な時は
何らかの**枝刈り**が必須

枝刈りがあると並列化は
難しくなる



単純に分割すると
計算負荷が不均等になり
大規模な並列化は不可能



並列 Monte-Carlo Tree Search

テクニカルすぎるので省略。鍵となるテクニックの名称は下図参照

並列囲碁プログラム mp-fuego (50 node vs 1 node で48勝2敗)

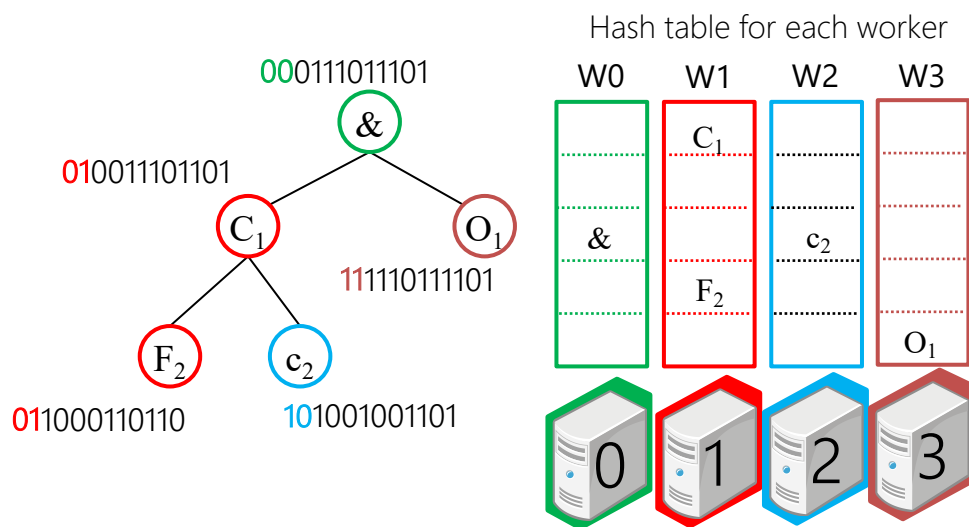
化合物探索 (逐次2560分 対 256並列10分がほぼ同性能)

実装は MPI_lprobe を使った技巧的なプログラムで、かなり複雑

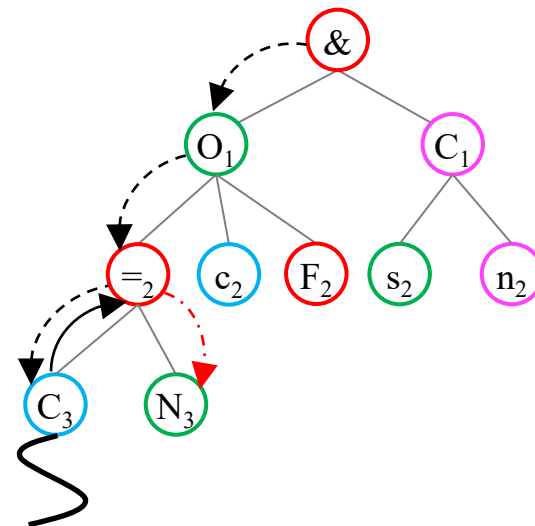
K. Yoshizoe, A. Kishimoto, T. Kaneko, H. Yoshimoto, Y. Ishikawa. "Scalable Distributed Monte-Carlo Tree Search." SoCS 2011.

X. Yang, T. Kr Aasawat, K. Yoshizoe. "Practical Massively Parallel Monte-Carlo Tree Search Applied to Molecular Design." ICLR 2021.

ハッシュ駆動並列化



深さ優先変形



ゲームAIの進歩の意義

囲碁・将棋・チェスを含む
多くのゲームで人類を超えた

機械学習 + 探索 + HPC

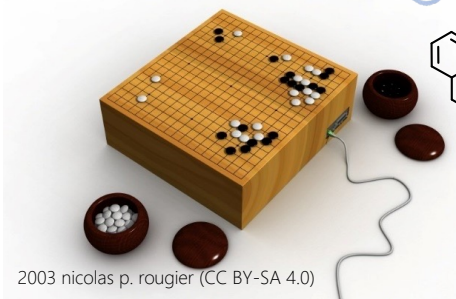
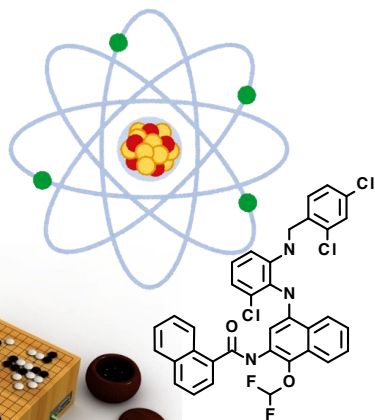
「完璧に精確なシミュレータ上で
計算実験をひたすら行うことで
より良い解を発見する」
というアプローチと一般化したい

これは強力なツールなので
ゲーム以外にも応用すべき

様々な数値シミュレーションの
技術を活用し、
多くの最適化問題を解きたい

DeepMind に機械学習メインでは
太刀打ちできないし...

手始めに化合物探索とか...



2003 nicolas p. rougier (CC BY-SA 4.0)

新規化合物探索

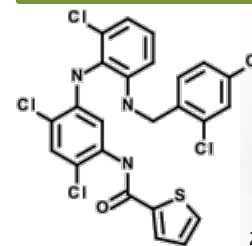
何らかの「スコア」が高い
化合物を発見したい

薬になりそうな
分子量の小さい物

チェス
 10^{45}



化合物
 10^{60}



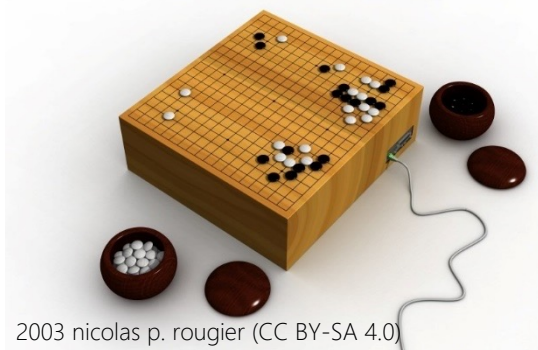
囲碁
 10^{170}



2003 nicolas p. rougier (CC BY-SA 4.0)

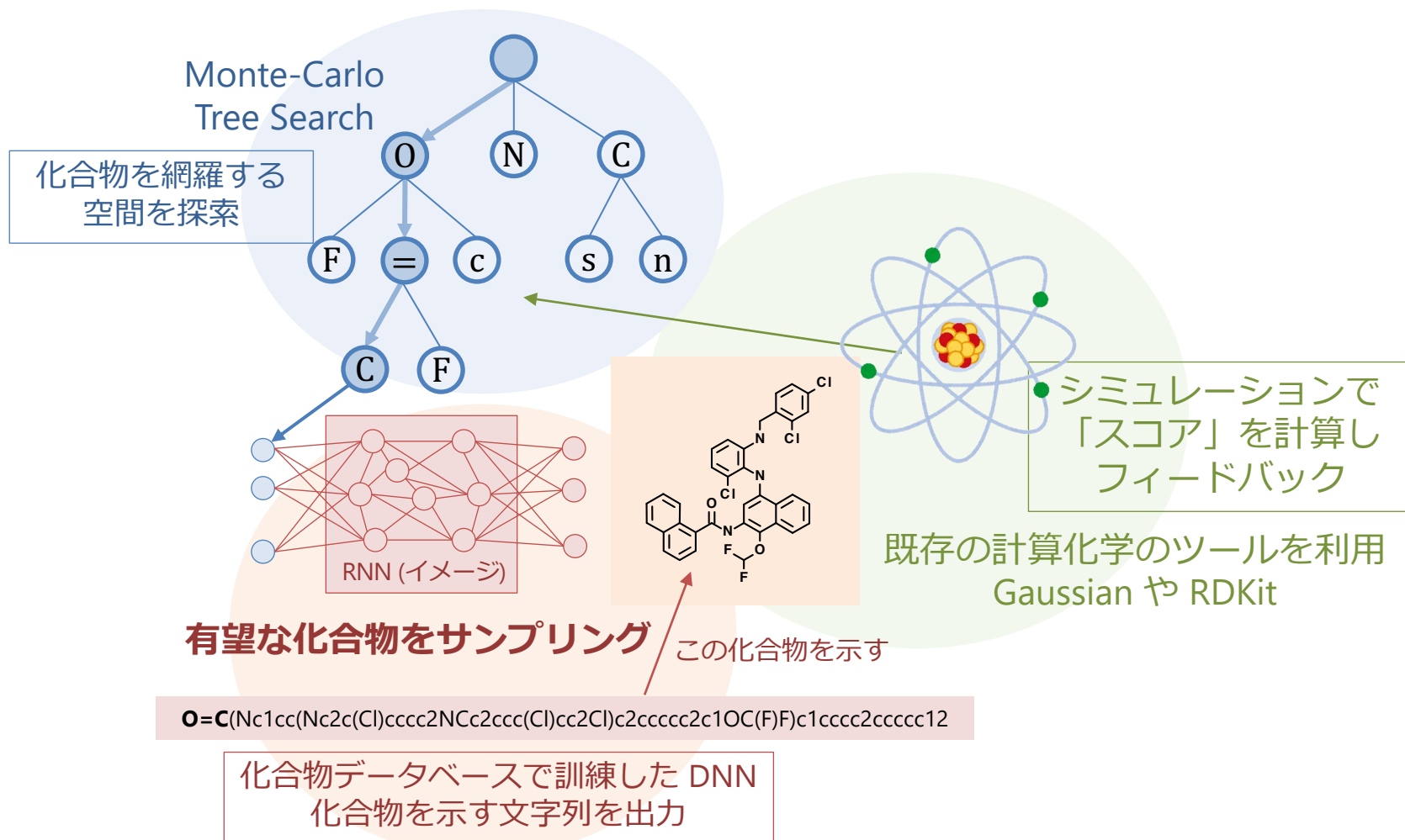
これを AlphaGo ぽい
手法で解く

全探索を行うには
探索空間が広すぎる



2003 nicolas p. rougier (CC BY-SA 4.0)

ChemTS: AlphaGo for Chemistry



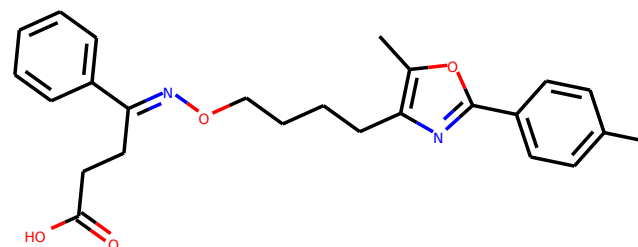
ChemTS: An Efficient Python Library for de novo Molecular Generation.
X. Yang, J. Zhang, **K. Yoshizoe**, K. Terayama, K. Tsuda. STAM 2017

Practical Massively Parallel Monte-Carlo Tree Search Applied to Molecular Design
X. Yang, T. Kr Aasawat, **K. Yoshizoe**. ICLR 2021

化合物の探索空間を文字列で表現

(SMILES: Simplified Molecular-Input Line-Entry System)

O	Water (H and single bond omitted)
O=C=O	Carbon dioxide
N#N	Nitrogen
c1=cc=cc=c1	Benzene (c1 and c1 connect)
[Cu+2].[O-]S(=O)(=O)[O-]	Copper sulfate



Cc3ccc(c2nc(CCCCO/N=C(CCC(O)=O)c1ccccc1)c(C)o2)cc3

以下の文法規則によって定義される

それぞれの記号は元素・結合・リングの情報を示す

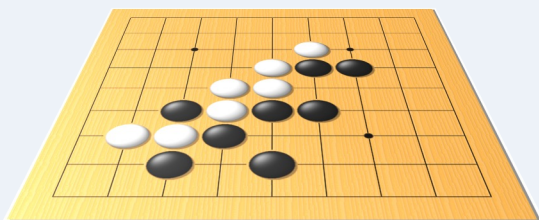
Atom: {C, c, o, O, N, F, [C@@H], n, -, S, Cl, [O-], [C@H], [NH+], [C@], s, Br, [nH], [NH3+], [NH2+], [C@@], [N+], [nH+], [S@], [N-], [n+], [S@@], [S-], I, [n-], P, [OH+], [NH-], [P@@H], [P@@], [PH2], [P@], [P+], [S+], [o+], [CH2-], [CH-], [SH+], [O+], [s+], [PH+], [PH], [S@@+]}

Bonds: {/, =, ¥# }

Ring: {1,2,3,4,5,6,7,8,9}

Branch: {(,)}

AlphaGo



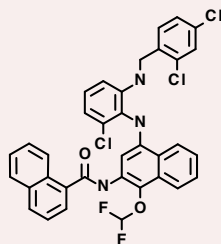
CNN (ResNet)
による画像認識

囲碁の局面のサンプリング
(rollout)

勝敗による報酬

Monte-Carlo Tree Search
(P-UCT)

ChemTS (我々の研究)



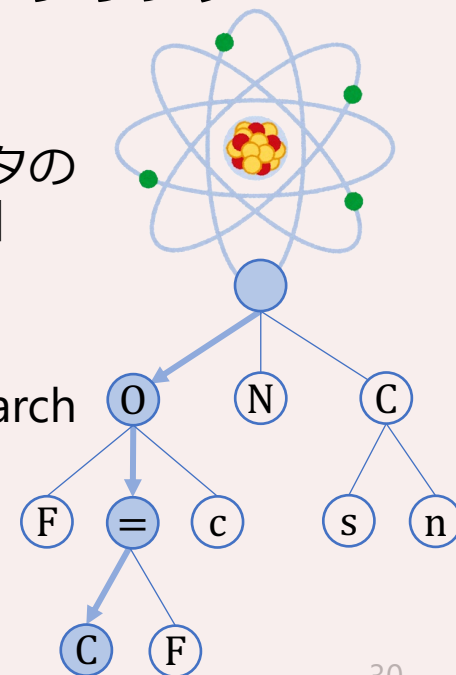
O=C(Nc1cc(Nc2c(Cl)cccc2NCc2ccc(Cl)cc2Cl)c2ccccc2c1OC(F)F)c1cccc2ccccc12

RNN (GRU) による文字列処理

RNN によるサンプリング
(rollout)

計算化学シミュレータの
結果に基づく報酬

Monte-Carlo Tree Search
(vanilla UCT)



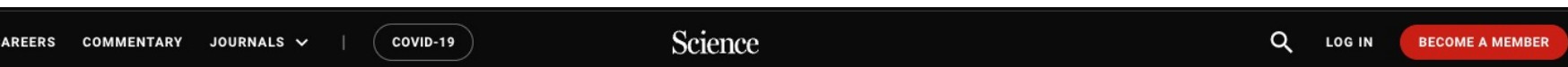
新規化合物の発見

並列探索で高速にスコアの高い化合物を発見 (ただし単純なベンチマークスコア)

X. Yang, T. Kr Aasawat, **K. Yoshizoe**. "Practical Massively Parallel Monte-Carlo Tree Search Applied to Molecular Design." ICLR 2021

並列探索で実際に新たな蛍光有機分子を発見 (これは共著者に頼り切りですが)
(Gaussian16 を用いた DFTシミュレーションによるスコアを利用)

M. Sumita, K. Terayama, N. Suzuki, S. Ishihara, R. Tamura, M. K. Chahal, D. T. Payne, **K. Yoshizoe**, K. Tsuda,
"De novo creation of a naked eye-detectable fluorescent molecule based on quantum chemical computation and machine learning",
Science Advances, 10.1126/sciadv.abj3906



ScienceAdvances

Current Issue

First release papers

Archive

About

Submit manuscript

GET OUR E-ALERTS



PHYSICAL AND MATERIALS SCIENCES | 9 MAR 2022

AI, quantum chemistry designed
fluorescent molecules



計算資源→ゲームの強さ→？名人

