Google Cloud

ニューラルネットワークは何を学んでいるのか？

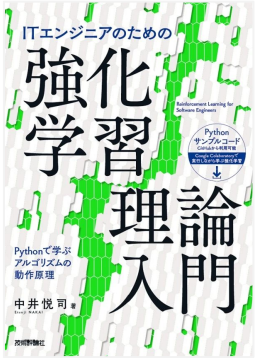
～ Explainable AI を応用したチェスエージェントの分析例

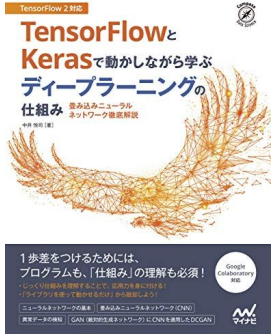
中井悦司 / Etsuji Nakai
Solutions Architect, Google Cloud

$ who am i

中井悦司 / Etsuji Nakai

Solutions Architect, Google Cloud





Google AI Blog

The latest from Google Research

Google Research: Themes from 2021 and Beyond

Tuesday, January 11, 2022

Posted by Jeff Dean, Senior Fellow and SVP of Google Research, on behalf of the entire Google Research community

Over the last several decades, I've witnessed a lot of change in the fields of machine learning (ML) and computer science. Early approaches, which often fell short, eventually gave rise to modern approaches that have been very successful. Following that long-arc pattern of progress, I think we'll see a number of exciting advances over the next several years, advances that will ultimately benefit the lives of billions of people with greater impact than ever before. In this post, I'll highlight five areas where ML is poised to have such impact. For each, I'll discuss related research (mostly from 2021) and the directions and progress we'll likely see in the next few years.

- Trend 1: More Capable, General-Purpose ML Models
- Trend 2: Continued Efficiency Improvements for ML
- Trend 3: ML Is Becoming More Personally and Communally Beneficial
- Trend 4: Growing Benefits of ML in Science, Health and
- Trend 5: Deeper and Broader Understanding of ML

より深く、より広く機械学習を理解する

https://ai.googleblog.com/2022/01/google-research-themes-from-2021-and.html

Acquisition of Chess Knowledge in AlphaZero

Thomas McGrath[1,+], Andrei Kapishnikov[2,+], Nenad Tomašev[1], Adam Pearce[2], Demis Hassabis[1], Been Kim[2], Ulrich Paquet[1], and Vladimir Kramnik[3]

[1]DeepMind
[2]Google Brain
[3]World Chess Champion, 2000–2007*
[+]these authors contributed equally to this work

ABSTRACT

What is learned by sophisticated neural network agents such as AlphaZero? This question is of both scientific and practical interest. If the representations of strong neural networks bear no resemblance to human concepts, our ability to understand faithful explanations of their decisions will be restricted, ultimately limiting what we can achieve with neural network interpretability. In this work we provide evidence that human knowledge is acquired by the AlphaZero neural network as it trains on the game of chess. By probing for a broad range of human chess concepts we show when and where these concepts are represented in the AlphaZero network. We also provide a behavioural analysis focusing on opening play, including qualitative analysis from chess Grandmaster Vladimir Kramnik. Finally, we carry out a preliminary investigation looking at the low-level details of AlphaZero's representations, and make the resulting behavioural and representational analyses available online.

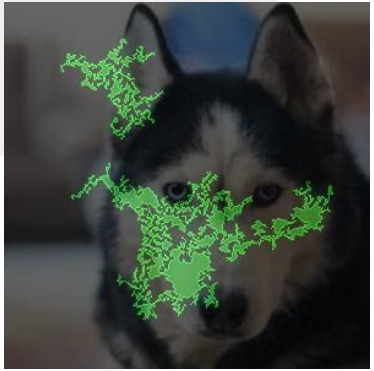
https://arxiv.org/abs/2111.09259

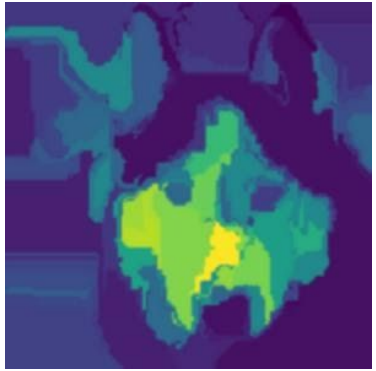
Explainable AI とは？

Explainable AI の例：画像分類モデル

- Vertex AI にデプロイしたカスタムモデルで利用可能
- 予測結果に与える影響が大きい領域を可視化

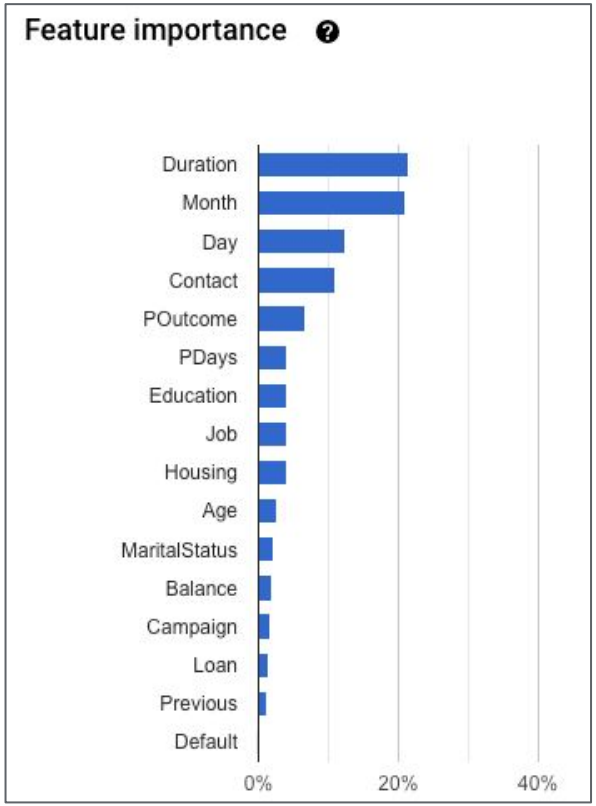
```
client = aip.PredictionServiceClient(client_options=client_options)
response = clients.explain(
        endpoint=endpoint,
        instances=instances,
        parameters=parameters,
        deployed_model_id=deployed_model_id,
)
```

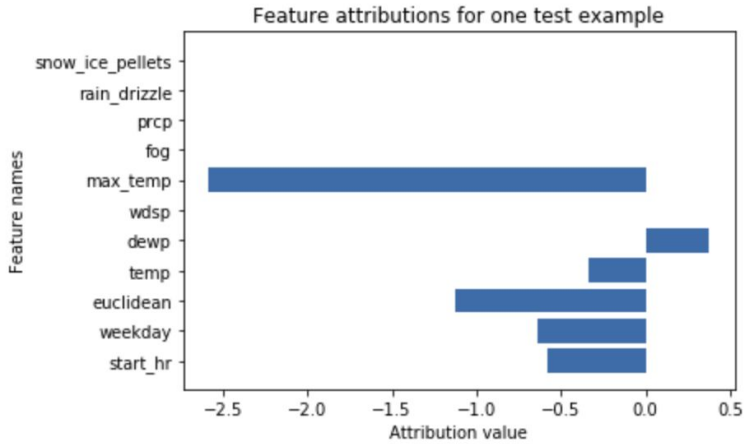




Explainable AI の例：AutoML Tables

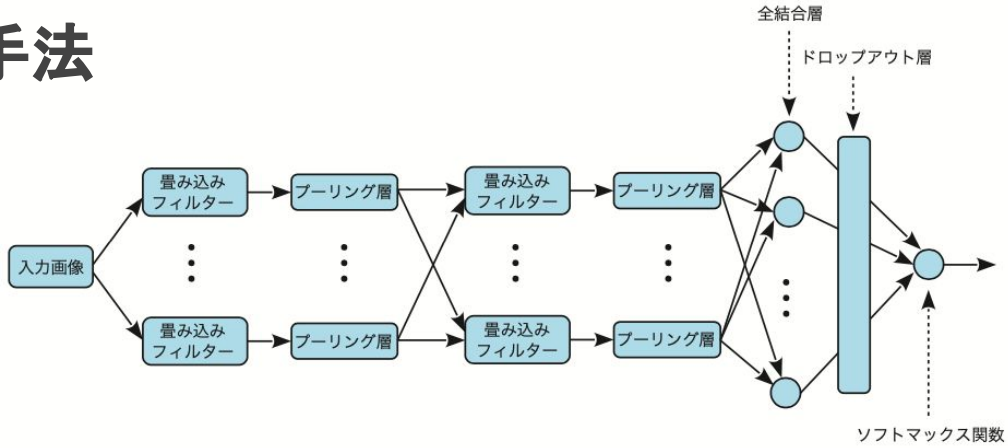
- モデル特徴量の重要度
  - モデル全体における、特徴量ごとの予測に与える影響の大きさ
- ローカル特徴量の重要度
  - 特定の予測結果における、特徴量ごとの影響度

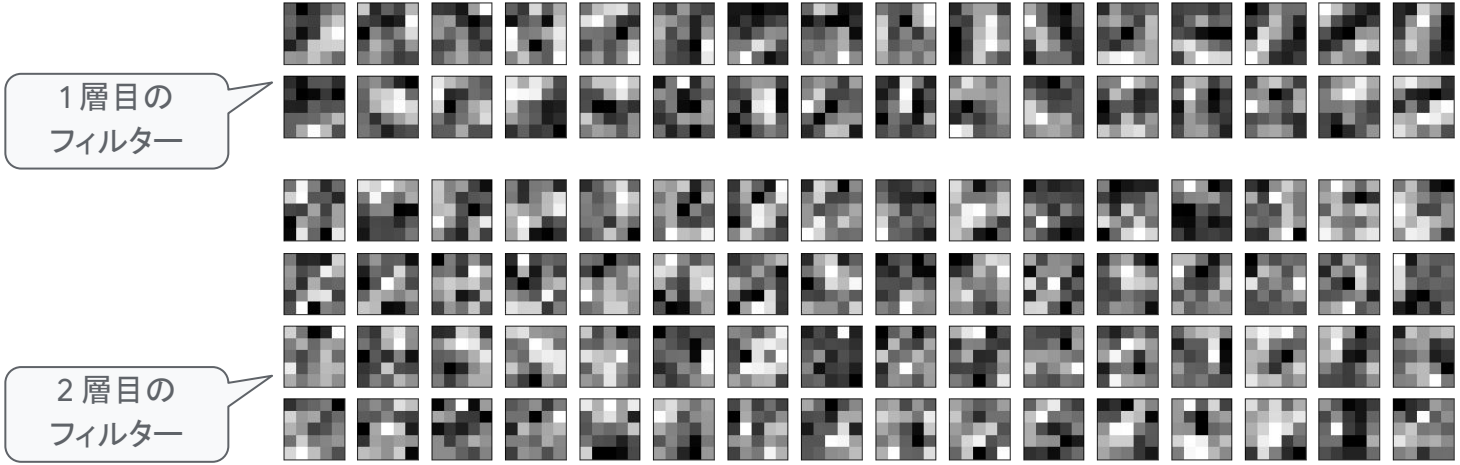




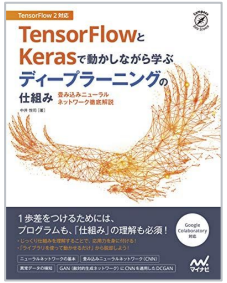
隠れ層の情報に着目した手法

- 学習後の畳み込みニューラルネットワーク（CNN）の畳み込みフィルターを可視化した様子



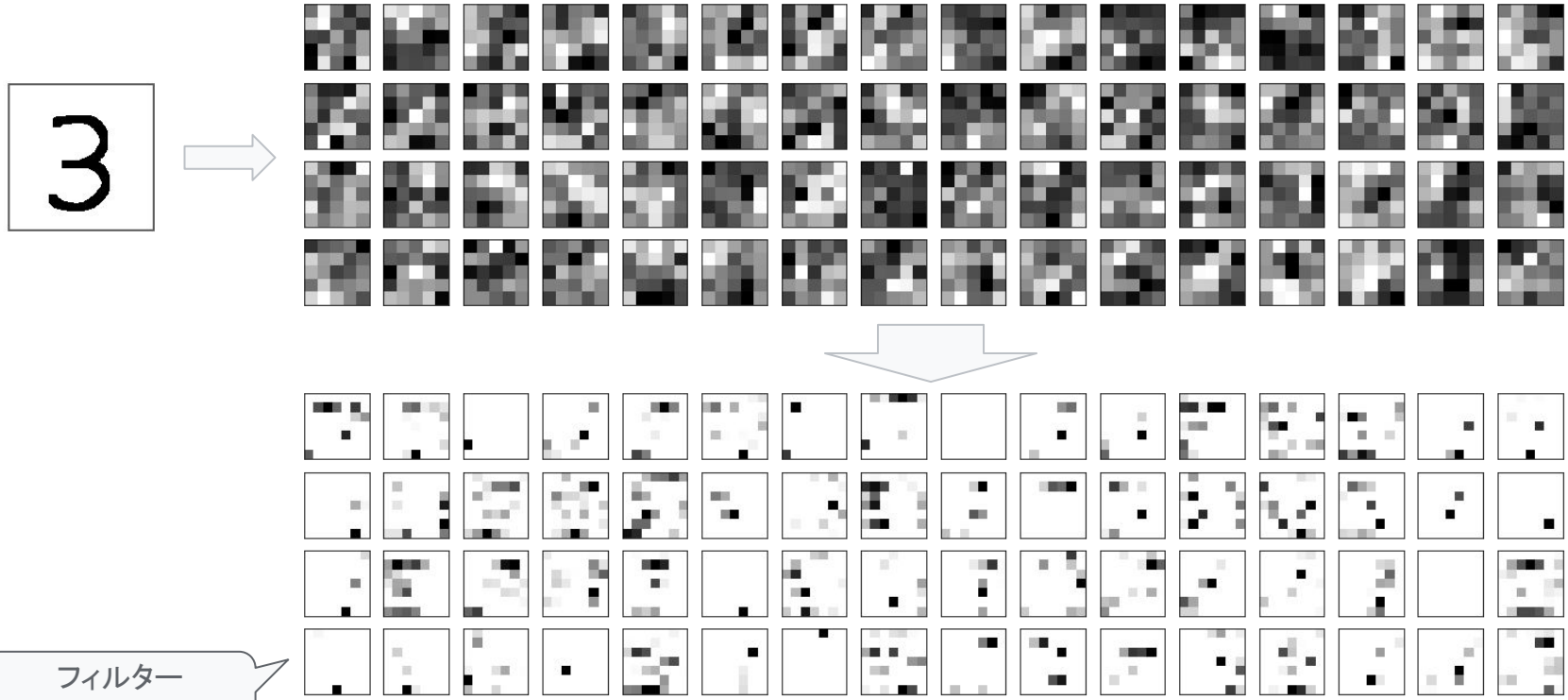


※こちらの書籍で
解説しています



隠れ層の情報に着目した手法

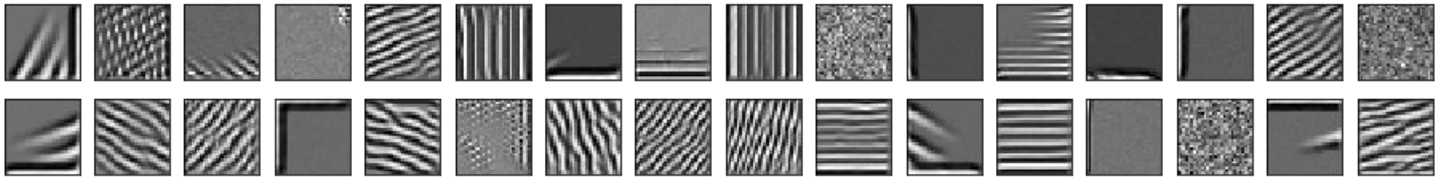
- 特定の画像を入力すると、フィルターによって出力の強弱があることがわかる



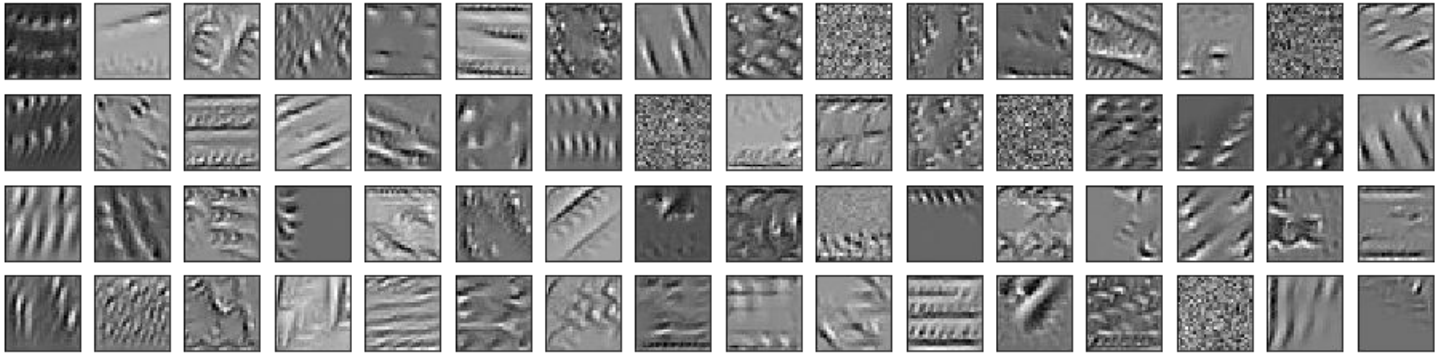
隠れ層の情報に着目した手法

- それぞれのフィルターについて、出力を最大化する入力画像を作成
  - 2 層目のフィルターの方がより複雑な図形パターンに反応していることがわかる

1 層目の
フィルターに対
する結果



2 層目の
フィルターに対
する結果



Acquisition of Chess Knowledge in AlphaZero

Thomas McGrath[1,+], Andrei Kapishnikov[2,+], Nenad Tomašev[1], Adam Pearce[2], Demis Hassabis[1], Been Kim[2], Ulrich Paquet[1], and Vladimir Kramnik[3]

[1]DeepMind
[2]Google Brain
[3]World Chess Champion, 2000–2007*
[+]these authors contributed equally to this work

ABSTRACT

What is learned by sophisticated neural network agents such as AlphaZero? This question is of both scientific and practical interest. If the representations of strong neural networks bear no resemblance to human concepts, our ability to understand faithful explanations of their decisions will be restricted, ultimately limiting what we can achieve with neural network interpretability. In this work we provide evidence that human knowledge is acquired by the AlphaZero neural network as it trains on the game of chess. By probing for a broad range of human chess concepts we show when and where these concepts are represented in the AlphaZero network. We also provide a behavioural analysis focusing on opening play, including qualitative analysis from chess Grandmaster Vladimir Kramnik. Finally, we carry out a preliminary investigation looking at the low-level details of AlphaZero's representations, and make the resulting behavioural and representational analyses available online.

https://arxiv.org/abs/2111.09259

論文解説

Acquisition of Chess Knowledge in AlphaZero

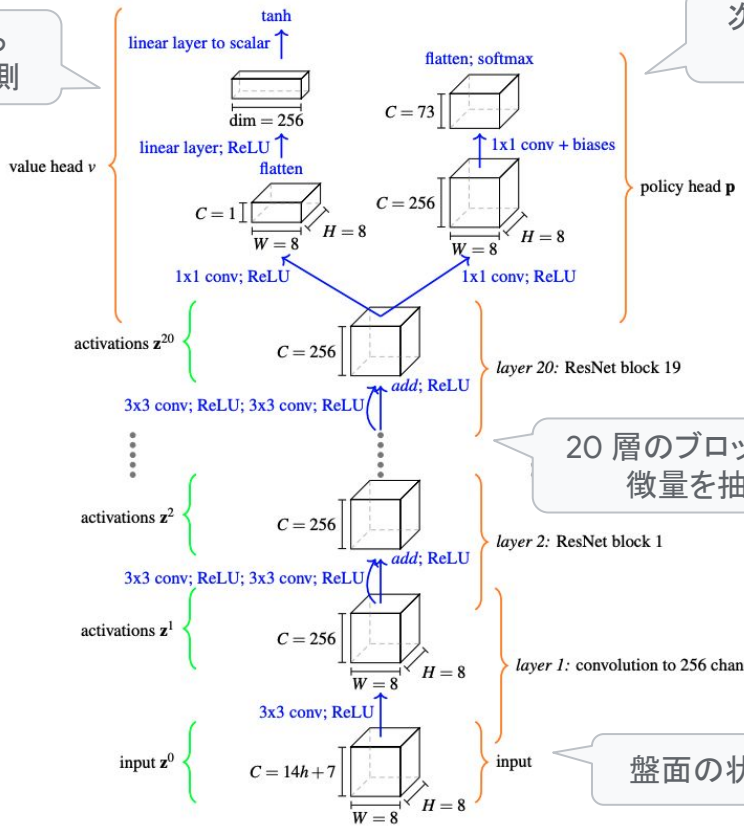
論文の概要

- Acquisition of Chess Knowledge in AlphaZero（2021 年）
  - AlphaZero : 人間の対局データを利用せずにAI 同士の自動対戦だけで学習したチェスのエージェントで、人間のプロ棋士を超える能力を持つ
- 次の 3 つの観点で AlphaZero のニューラルネットワークの構造を調査
  - 学習後のニューラルネットワーク内に、人間のプロ棋士の「考え方」を特徴量化する機能が存在するか検証
  - ニューラルネットワークの各ブロックの出力に教師なし学習を適用して、どのような情報を表現しているかを探索（人間とは異なる「考え方」が存在するかを検証）
  - 学習中に打ち手の傾向がどのように変化したかを確認し、歴史的な傾向の変化と比較

AlphaZero のネットワーク構造

現在の状況から
勝てる確率を予測

次に打つべき手
（確率）を予測



畳み込みフィルターを持つ同じ構造の
ブロックが 20 回繰り返される

20 層のブロックで特
徴量を抽出

盤面の状態を入力

AlphaZero のネットワーク構造

- 入力データ
  - 過去 8 手分を含む 8 × 8 の盤面の情報
  - 6 種類の駒それぞれの配置、何手目の状態か、「キャスリング」の可否などの情報を8 × 8 × 119 サイズのリストに記録
- 隠れ層のブロック（ResNet Block）
  - 畳み込みフィルターを持つ同じ構造のブロックが20 回繰り返される
  - 各ブロックの出力はすべて同じ8 × 8 × 256 サイズ

それぞれのブロックはどのような情報を抽出しているのか？

- 出力データ
  - Value head：盤面の価値（盤面がどの程度有利な状態か）
  - Policy head：次に打つべき手の確率分布（順位付きリスト）

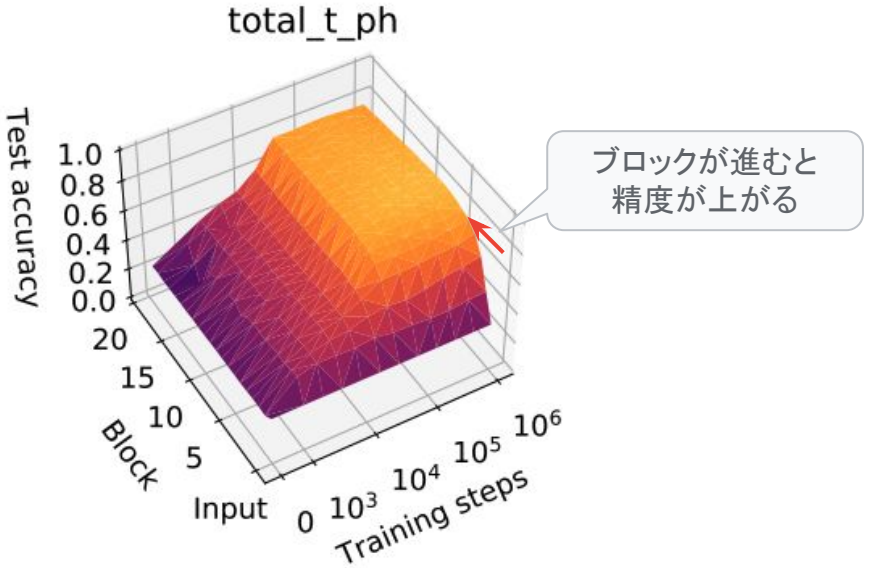
人間の「考え方」を特徴量化しているか？

人間の「考え方」を特徴量化しているか？

- 人間の「考え方」とは？
  - 「Stockfish（ストックフィッシュ）の評価関数」をモデルとして利用
    → 与えられた盤面に対して、一定のルールで複数の「スコア」を計算
  - material（それぞれの駒の位置）、imbalance（持ち駒の差異）、mobility（駒の動きやすさ）、king_safety（キングの安全性）などの要素を個別に点数化
  - プロ棋士が盤面を評価する際に用いる主要な評価基準に対応すると考えられる
- AlphaZero の特徴量との関係
  - 各ブロックの出力値の単純な線形関数で評価関数の出力値を再現できるかを確認
    → 評価関数の値を正解ラベルとする学習データで教師あり学習を実施
    入力値：盤面、モデル：ブロックの出力値の線形関数、正解ラベル：評価関数の出力値
  - 再現できる場合、そのブロックの特徴量は該当の「考え方」を内包していると考えられる

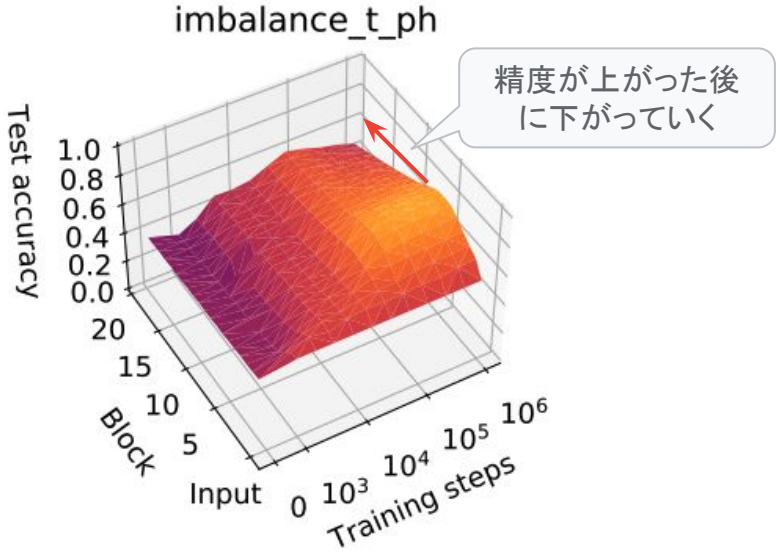
人間の「考え方」を特徴量化しているか？

代表的な指標
の総合評価



(a) Stockfish 8's total score

持ち駒の差異
による評価



(h) Past 10^5 training steps, StockFish 8's material imbalance score becomes *less* predictable from AlphaZero's later layers.

※ すべてのスコアの説明と学習結果が論文に掲載

Concept names	Description
pawn_fork [m|o]	True if a pawn is attacking two pieces of higher value (knight, bishop, rook, queen, or king) and is not pinned.
knight_fork [m|o]	True if a knight is attacking two pieces of higher value (rook, queen, or king) and is not pinned.
bishop_fork [m|o]	True if a bishop is attacking two pieces of higher value (rook, queen, or king) and is not pinned.
rook_fork [m|o]	True if a rook is attacking two pieces of higher value (queen, or king) and is not pinned.
has_pinned_pawn [m|o]	True if the side has a pawn that is pinned to the king of the same colour.
has_pinned_knight [m|o]	True if the side has a knight that is pinned to the king of the same colour.
has_pinned_bishop [m|o]	True if the side has a bishop that is pinned to the king of the same colour.
has_pinned_rook [m|o]	True if the side has a rook that is pinned to the king of the same colour.
has_pinned_queen [m|o]	True if the side has a queen that is pinned to the king of the same colour.
material [m|o|diff]	Material calculated as (#♙) + 3 * (#♘) + 3 * (#♗) + 5 * (#♖) + 9 * (#♕)
num_pieces [m|o|diff]	Number of pieces that a side has.
in_check	True if the side that makes a turn is in check.
has_bishop_pair [m|o]	True if the side has a pair of bishops.
has_connected_rooks [m|o]	True if the side has connected rooks.
has_control_of_open_file [m|o]	True if the side controls an open file (with the rooks, queen)
has_mate_threat	True if the opponent could mate the current side in a single move if the turn was passed to the opponent.
has_check_move [m|o]	True if the side can check the opponent's King.
can_capture_queen [m|o]	True if the side can capture the opponent's queen.
num_king_attacked_squares [m|o|diff]	The number of squares around the opponent's king that the playing side attacks. Can include occupied squares.
has_contested_open_file	True if an open file is occupied simultaneously by a rook and/or queen of both colours.
has_right_bc_ha_promotion [m|o]	True if 1) the side has a passed pawn on a or h files and 2) the side has a bishop that is of the colour of the promotion square of that pawn.
num_scb_pawns_same_side [m|o|diff]	The number of own pawns that occupy squares of the same colour as the colour of own bishop. Applicable only when the side has a single bishop.
num_ocb_pawns_same_side [m|o|diff]	The number of own pawns that occupy squares of the opposite colour to that of own bishop. Applicable only when the side has a single bishop.
num_scb_pawns_other_side [m|o|diff]	The number of opponent's pawns that occupy the squares of the same colour as the colour of own bishop. Applicable only when the side has a single bishop.
num_ocb_pawns_other_side [m|o|diff]	The number of opponent's pawns that occupy the squares of the opposite colour to the colour of own bishop. Applicable only when the side has a single bishop.
capture_possible_on_{sq} [m|o] sq=[d1|d2|d3|e1|e2|e3|g5|b5]	True is the side can capture a piece on the given square. The squares are named as if the side were playing White.
capture_happens_next_move_... ...on_{sq} sq=[d1|d2|d3|e1|e2|e3|g5|b5]	True if the capture of a piece on the given square had happened according to the game data. The squares are named as if the side were playing White.

Table 3. Custom chess concepts (self implemented; i.e. not from Stockfish 8's API) used in this paper. We use `m` as shorthand for `mine` and `o` as shorthand for `opponent`. `diff` stands for the `difference` between the mine and opponent values of the same concept.

35/69

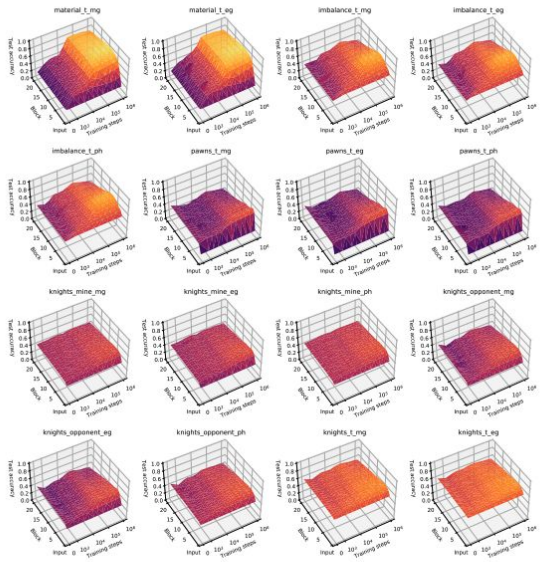


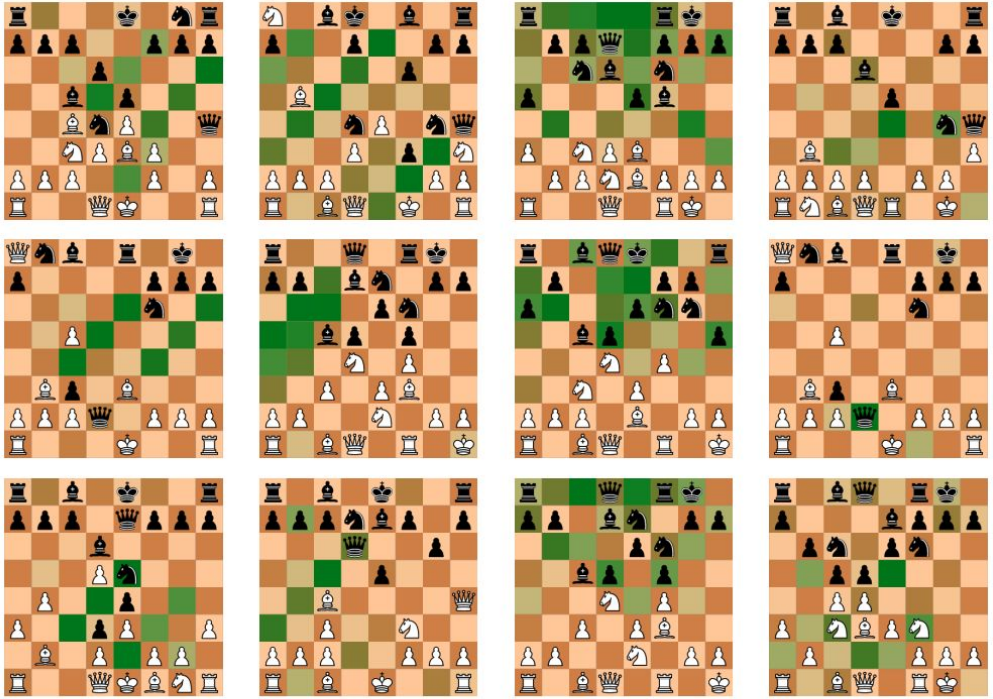
Figure 14. Regression results for Stockfish concepts from Table 2.

37/69

各ブロックの出力は
どのような情報を表現しているか？

教師なし学習による特徴量の抽出

- ブロックごとに（出力値の集合データに対する）Matrix Factorization を適用して、ブロックの出力を主要コンポーネントに分解
- 各主要コンポーネントが盤面のどの位置の情報を抽出しているかを表示（特定の盤面について、主要コンポーネントに対する重みを元の盤面にオーバーレイで表示）



(a) Development of diagonal moves for player (block 1, factor 26 of 36).

(b) Fully developed diagonal moves for opponent (block 3, factor 22 of 36).

(c) Count of opponent's potential piece moves (block 3, factor 11 of 36).

(d) Potential good squares to move to? (block 18, factor 22 of 36).

（参考）Matrix Factorization の考え方

Z = FG

- 映画のレーティングデータから映画の「隠れたグループ」が発見できる
- 各ユーザーは、グループ内の映画に対して類似した評価を与えており、同じグループの映画には、何らかの類似性があると想像される

Z	映画 A	映画 B	映画 C	映画 D	映画 E	映画 F	映画 G
ユーザー A	5	1	5	5	1	2	2
ユーザー B	1	3	1	1	3	1	1
ユーザー C	2	1	2	2	1	4	4
ユーザー D	5	4	5	5	4	2	2
ユーザー E	3	2	3	3	2	5	5

F	グループ 1	グループ 2	グループ 3
ユーザー A	5	1	2
ユーザー B	1	3	1
ユーザー C	2	1	4
ユーザー D	5	4	2
ユーザー E	3	2	5

グループに対する評価

「隠れたグループ」

G	映画 A	映画 B	映画 C	映画 D	映画 E	映画 F	映画 G
グループ 1	1		1	1			
グループ 2		1			1		
グループ 3						1	1

Matrix Factorization の適用方法

- モデルの入力（＝盤面の状態）
  - レーティングを与える「ユーザー」に対応させる
- 各ブロックの出力：8 × 8 × 256 サイズ
  - 最後の 256 を「256 種類の映画」に対応させる
  - 最初の 8 × 8 を各映画に対する「レーティング（評価）」と解釈

つまり、各ブロックは、1 つの盤面に対して 256 種類の観点による評価を与える

- さらに、256 種類の観点による評価を Matrix Factorization で 36 種類のグループにまとめる
  - 類似の観点をグループ化することで、「観点」の本質を抽出して解釈を容易にする
  - 各評価は 8 × 8 のデータであり「該当の評価基準における各マス目に対する注目度」と解釈できる

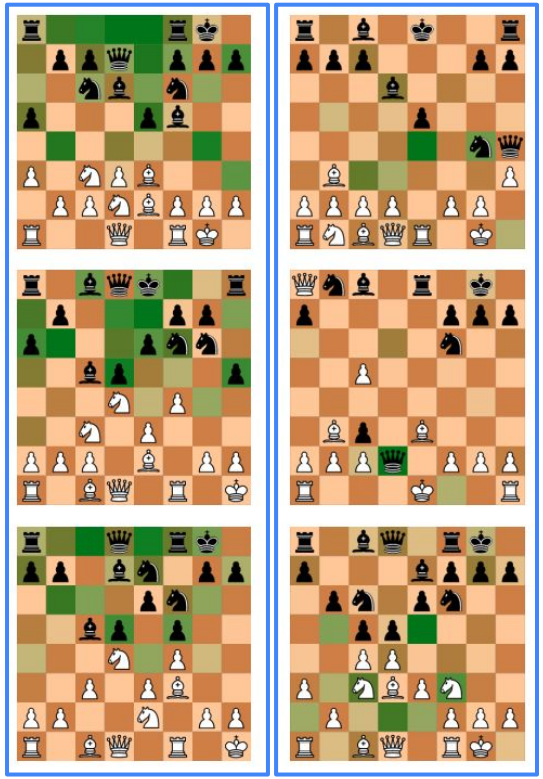
最終的に、各ブロックからは、与えられた盤面に対する 36 種類の評価（各マス目に対する注目度）が得られる

Matrix Factorization の適用結果

- 左側：3 種類の盤面について、3 番目のブロックの評価項目の 1 つ（8 × 8 のデータ）を入力盤面に重ねて表示
  - 「相手プレイヤーが次に打ちそうなマス」に対応していそうな雰囲気
- 右側：3 種類の盤面について、18 番目のブロックの評価項目の 1 つ（8 × 8 のデータ）を入力盤面に重ねて表示
  - 「次に打つと良さそうな手の候補」を示していそうな雰囲気
- 前段のブロックで基本的な情報を抽出した後に、それらを用いて、後段のブロックでより高度な情報を抽出しているように思われる

3 番目のブロックの評価項目の 1 つ

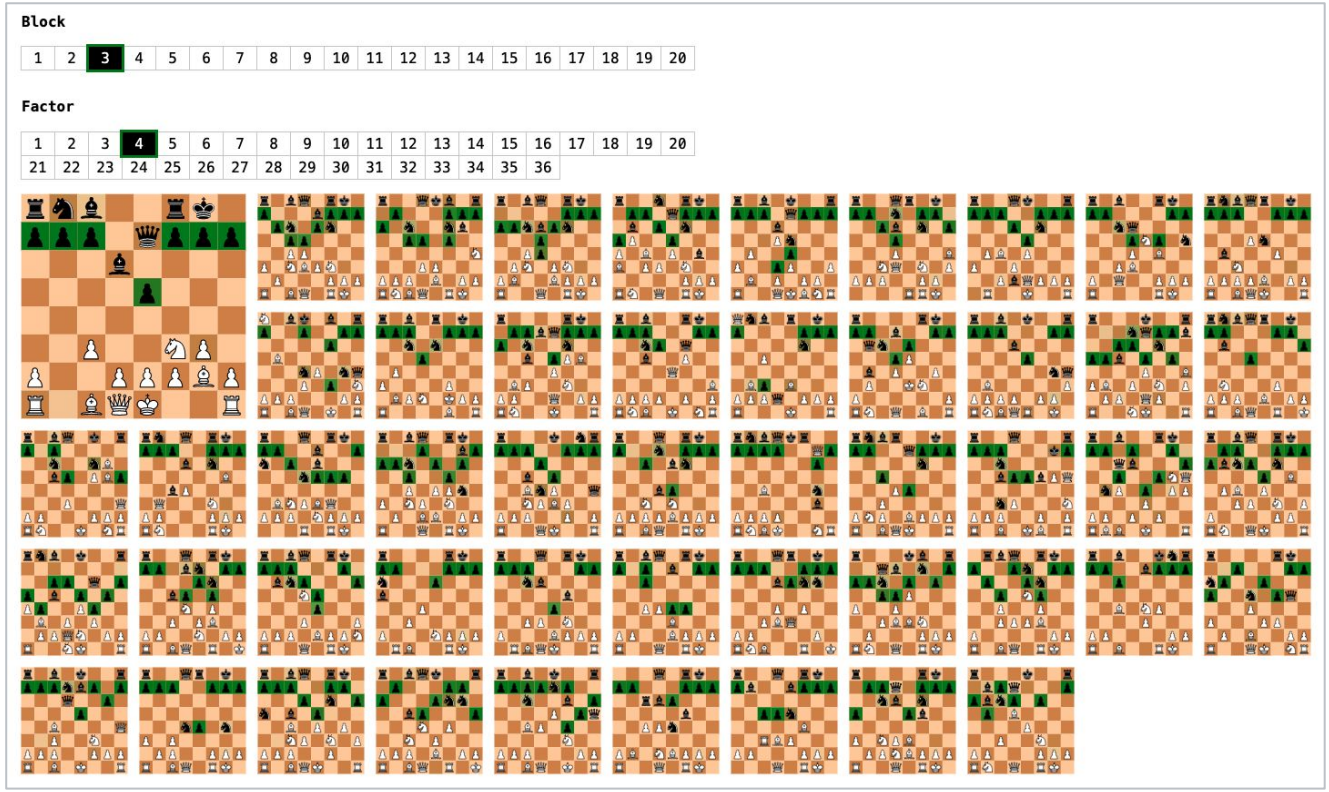
18 番目のブロックの評価項目の 1 つ



(c) Count of opponent's potential piece moves (block 3, factor 11 of 36).

(d) Potential good squares to move to? (block 18, factor 22 of 36).

※ すべての評価結果をWeb サイトで公開

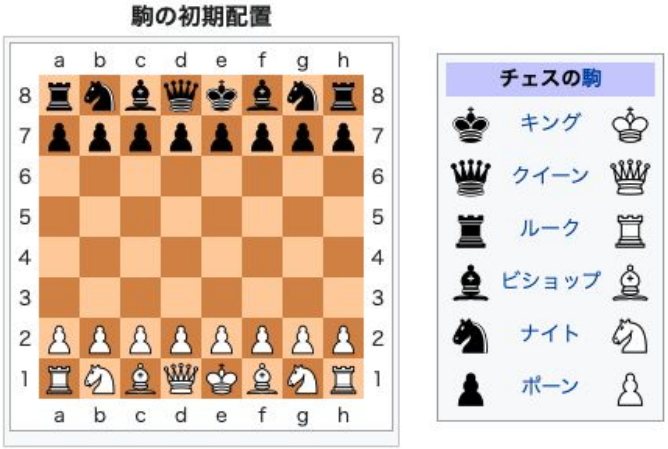


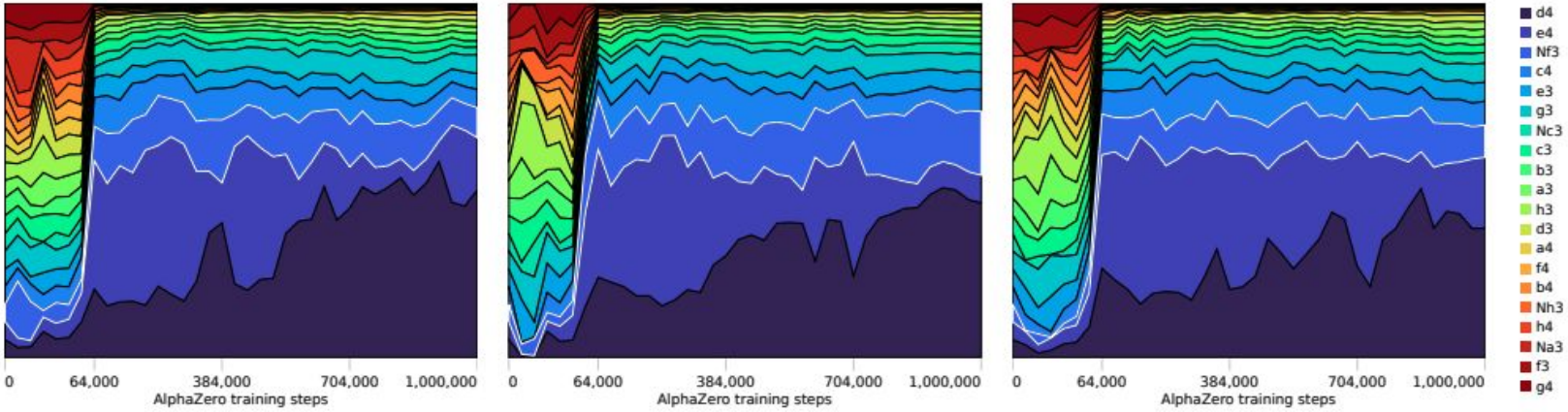
https://storage.googleapis.com/uncertainty-over-space/alphachess/index.html

学習中の打ち手の傾向変化を
チェスの歴史と比較

学習中の「初手」の傾向変化

- 学習初期は可能な手をほぼランダムに選択
- 64,000 ステップ辺りで急激に変化
- その後の学習で、頻度の高い手が e4 から d4 へと変化
- その他には、Nf3, c4 なども選択





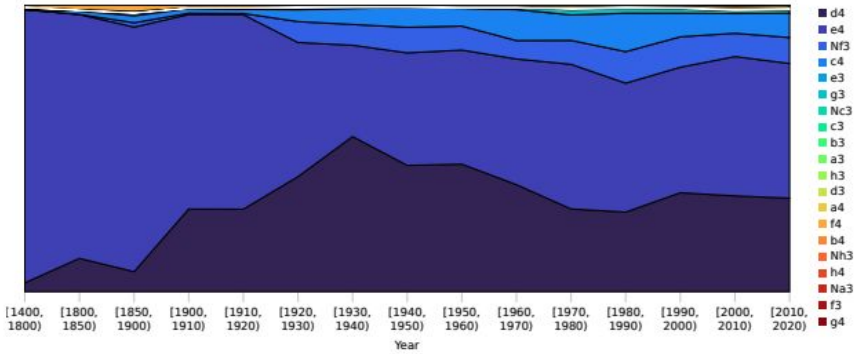
(b) The AlphaZero policy head's preferences of opening move, as a function of training steps. Here AlphaZero was trained three times from three different random seeds. AlphaZero's opening evolution starts by weighing all moves equally, no matter how bad, and then narrows down options. It stands in contrast with the progression of human knowledge, which gradually expanded from 1. e4.

学習中の「初手」の傾向変化

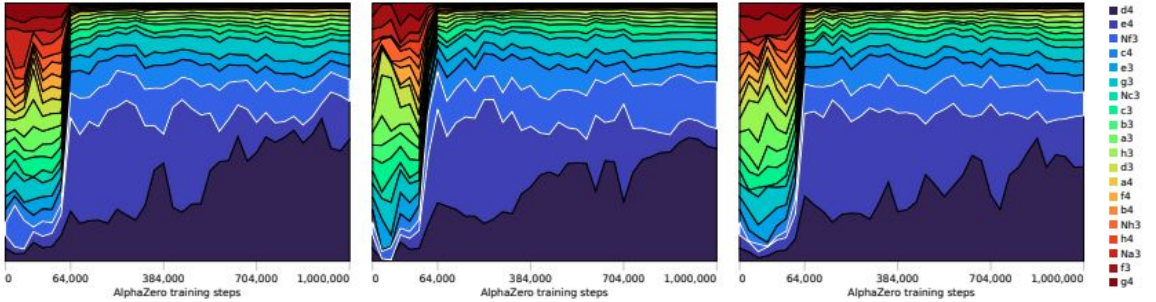
「初手」の傾向の
歴史的変化

- チェスの歴史の初期は、e4 が定番の選択
- その後、d4 が発見される
- 現代では、d4, e4, Nf3, c4 が主な選択肢
  → AlphaZero と同じ傾向

- AlphaZero は積極的な探索により、学習の初期から広い範囲の手を発見していることがわかる



(a) The evolution of the first move preference for White over the course of human history, spanning back to the earliest recorded games of modern chess in the Chessbase database. The early popularity of 1. e4 gives way to a more balanced exploration of different opening systems and an increasing adoption of more flexible systems in modern times.



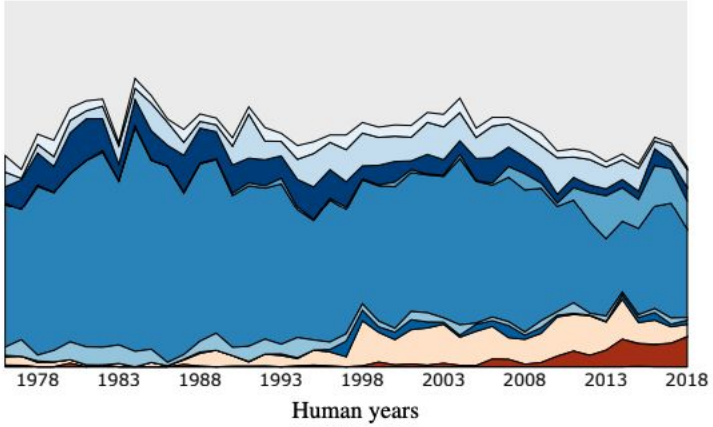
(b) The AlphaZero policy head's preferences of opening move, as a function of training steps. Here AlphaZero was trained three times from three different random seeds. AlphaZero's opening evolution starts by weighing all moves equally, no matter how bad, and then narrows down options. It stands in contrast with the progression of human knowledge, which gradually expanded from 1. e4.

ルイ・ロペスからの展開

- ルイ・ロペス：チェスの歴史が始まってから現代まで、数多く指されている定番のオープンニング
  - 先手が 3 手目を打つまでの「1. e4 e5 2. Nf3 Nc6 3. Bb5」という流れ
  - 後手の 3 手目から多くの変化に分かれる

ルイ・ロペス

過去30年間にグランドマスター（チェス棋士の最高位タイトル保持者）が最も良く打ったパターン

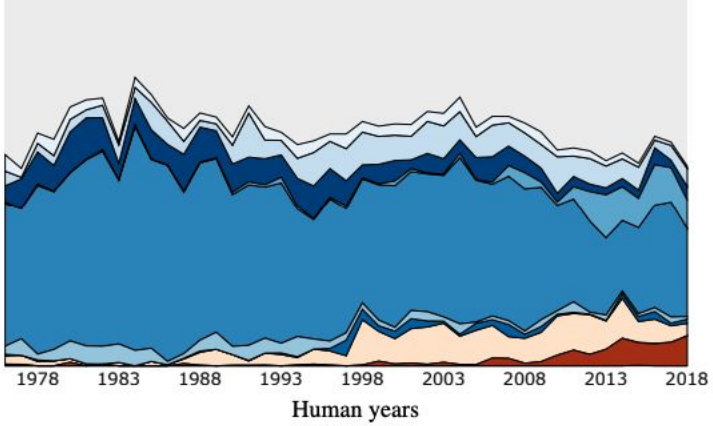


後手の 3 手目以降の展開パターン（ツリー図）

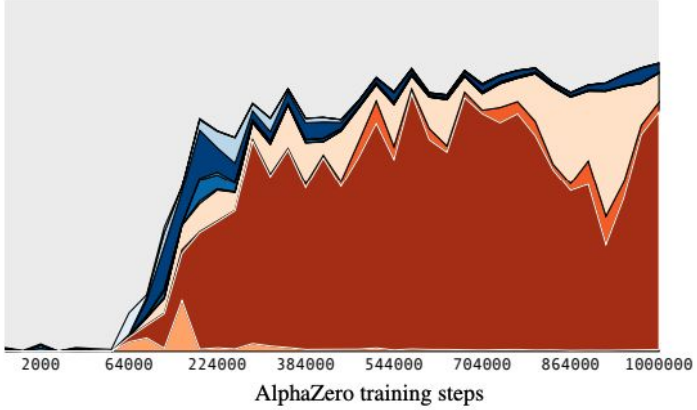
ルイ・ロペスからの展開

- AlphaZero は学習の初期段階でベルリン・ディフェンスを発見して、積極的に活用している
- 人間のプレイヤーはそれぞれの時代における「常識」に縛られる一方、 AlphaZeroは常識に縛られることなく、合理的に学習を進めているとも想像される

過去30年間にグランドマスター（チェス棋士の最高位タイトル保持者）が最も良く打ったパターン



学習中の AlphaZero がよく選択するパターン



まとめ

- Acquisition of Chess Knowledge in AlphaZero（2021 年）
  - AlphaZero : 人間の対局データを利用せずにAI 同士の自動対戦だけで学習したチェスのエージェントで、人間のプロ棋士を超える能力を持つ
- 次の 3 つの観点で AlphaZero のニューラルネットワークの構造を調査
  - 学習後のニューラルネットワーク内に、人間のプロ棋士の「考え方」を特徴量化する機能が存在するか検証
  - ニューラルネットワークの各ブロックの出力に教師なし学習を適用して、どのような情報を表現しているかを探索（人間とは異なる「考え方」が存在するかを検証）
  - 学習中に打ち手の傾向がどのように変化したかを確認し、歴史的な傾向の変化と比較

Thank you.

Google Cloud