

Beating Humans in complex Boardgames

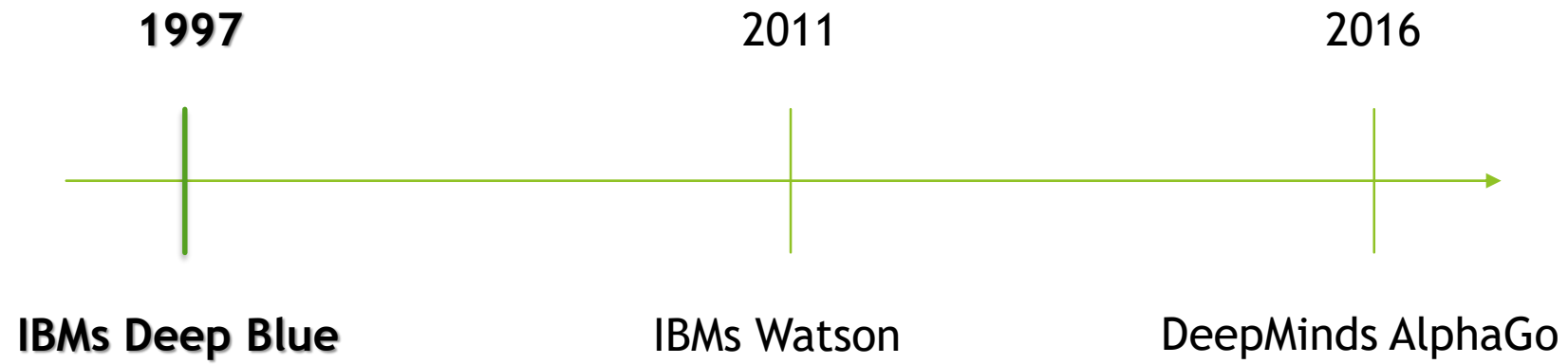
Seminar „Neueste Trends in Big Data Analytics“ - Universität Hamburg

Eike Nils Knopp - 18.12.2017

Inhalt

- ▶ Historie
 - ▶ IBMs Deep Blue
 - ▶ IBMs Watson
 - ▶ DeepMinds AlphaGO
- ▶ Warum ist AlphaGo so besonders?
- ▶ Hauptteil
 - ▶ Traditioneller Ansatz für Spiele-Als
 - ▶ Funktionsweise Deep Blue
 - ▶ Unterschiede Schach und Go
 - ▶ Funktionsweise Alpha Go
- ▶ Ausblick
 - ▶ Wie kann das neue Wissen genutzt werden

Historie

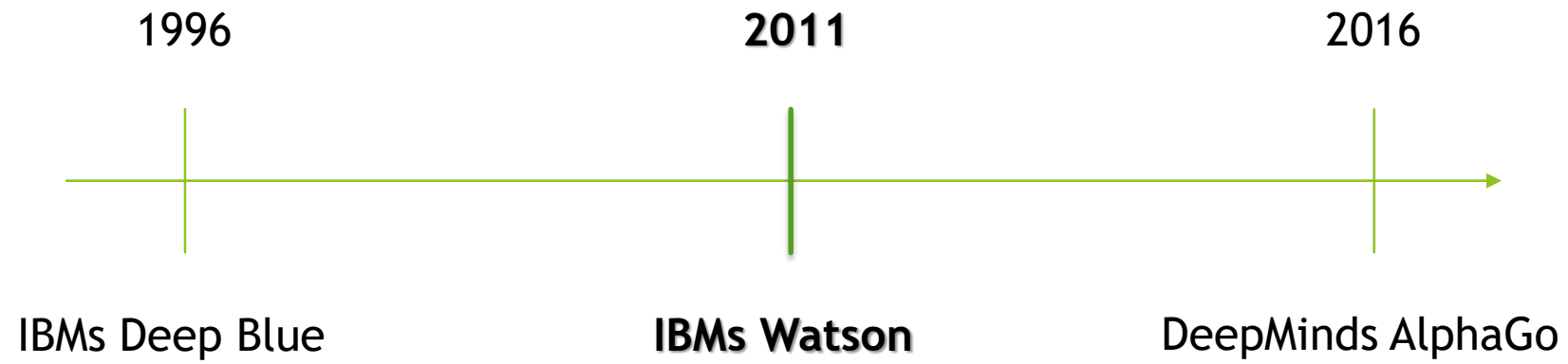


IBMs Deep Blue (1997)



- ▶ Schlug Schachweltmeister Garry Kasparov
- ▶ IBM siegt $3\frac{1}{2}$ zu $2\frac{1}{2}$
- ▶ Kombination TreeSearch + Evaluate-Funktion
- ▶ Erster Sieg gegen Weltmeister

Historie



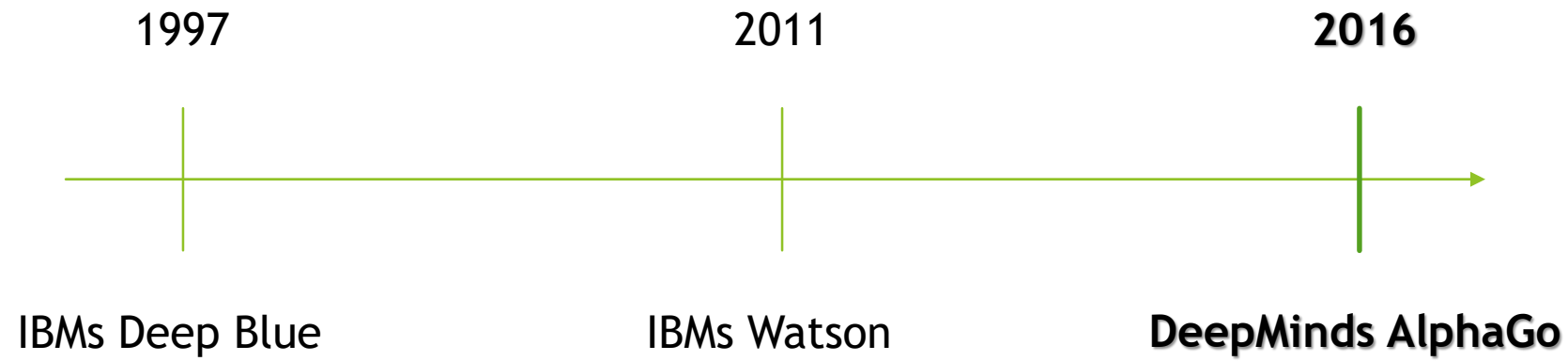
IBMs Watson (2011)



[2] <https://sciencebasedmedicine.org/tag/watson/>

- ▶ Schlug gleich mehrere Champions gleichzeitig
- ▶ Verstehen von Fragen in natürlicher Sprache
- ▶ Deep analytics und natural language processing
- ▶ Antwortet basierend auf Confidence-Level

Historie



Deep Minds AlphaGo (2016)

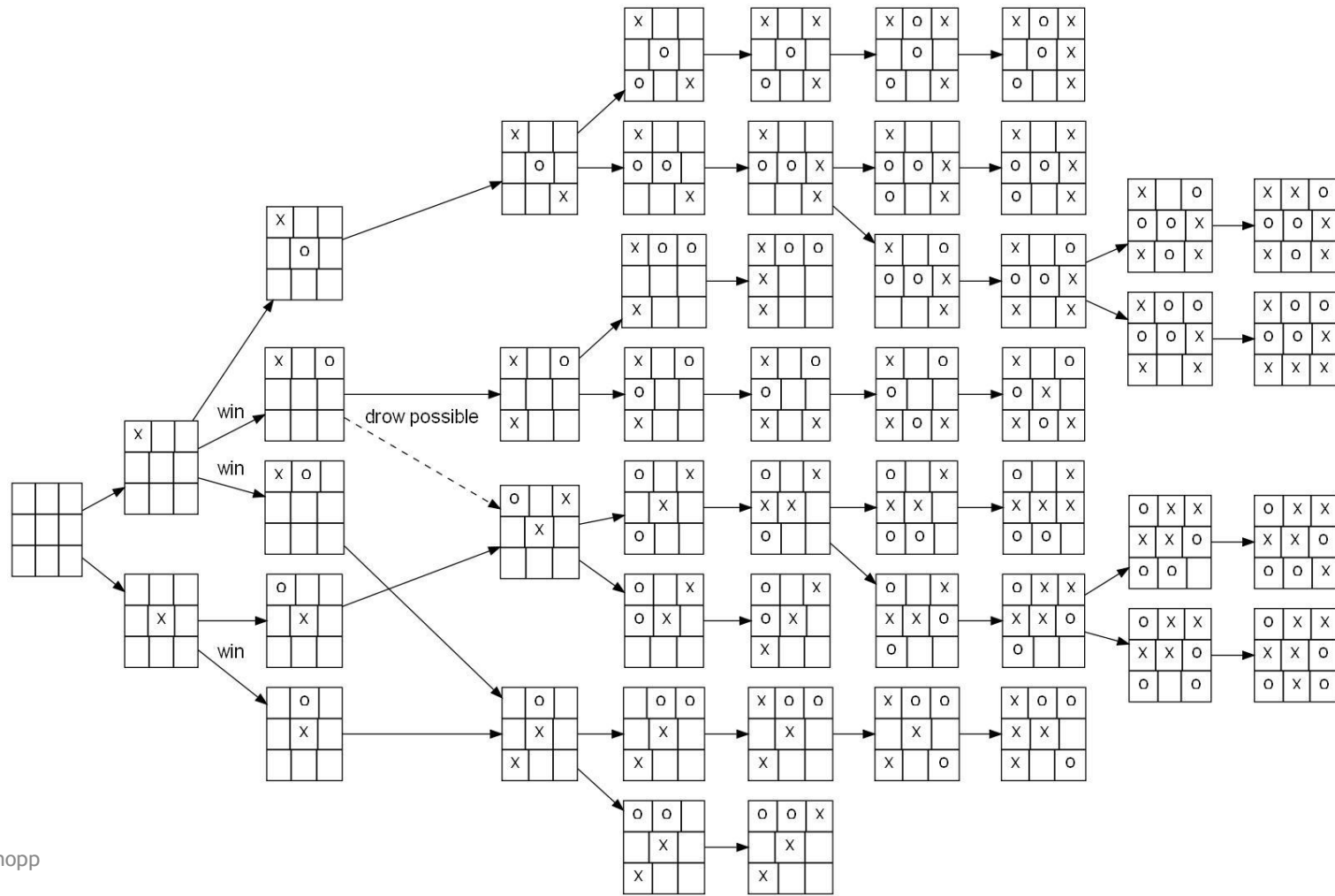


- ▶ Schlug 18-fachen Weltmeister Lee Sedol
- ▶ Klarer Sieg mit 4 - 1
- ▶ Erlangte höchstmöglichen Go-Rang

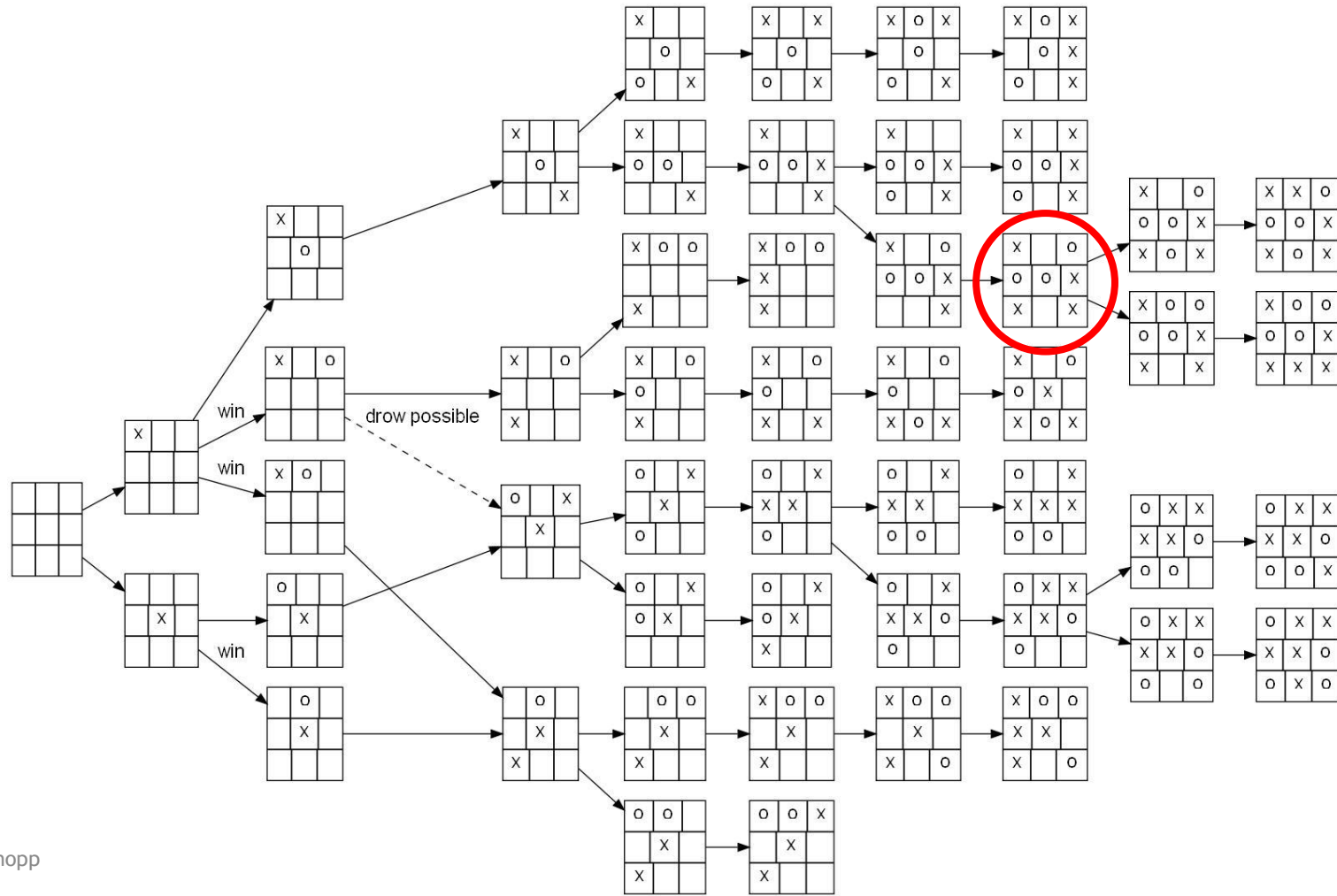
Relevanz von AlphaGo

Warum ist AlphaGo so besonders?

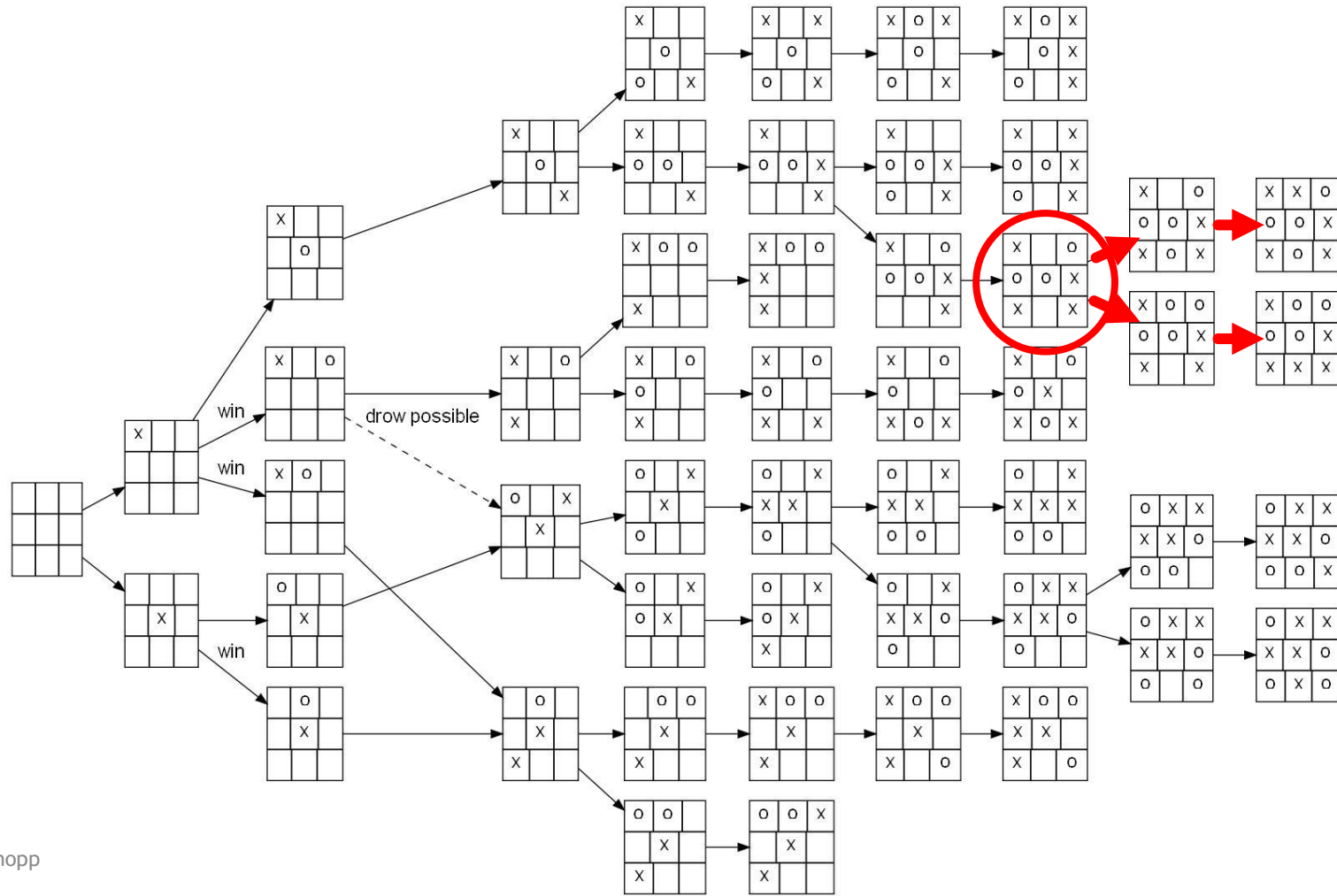
Traditioneller Ansatz für Brettspiele



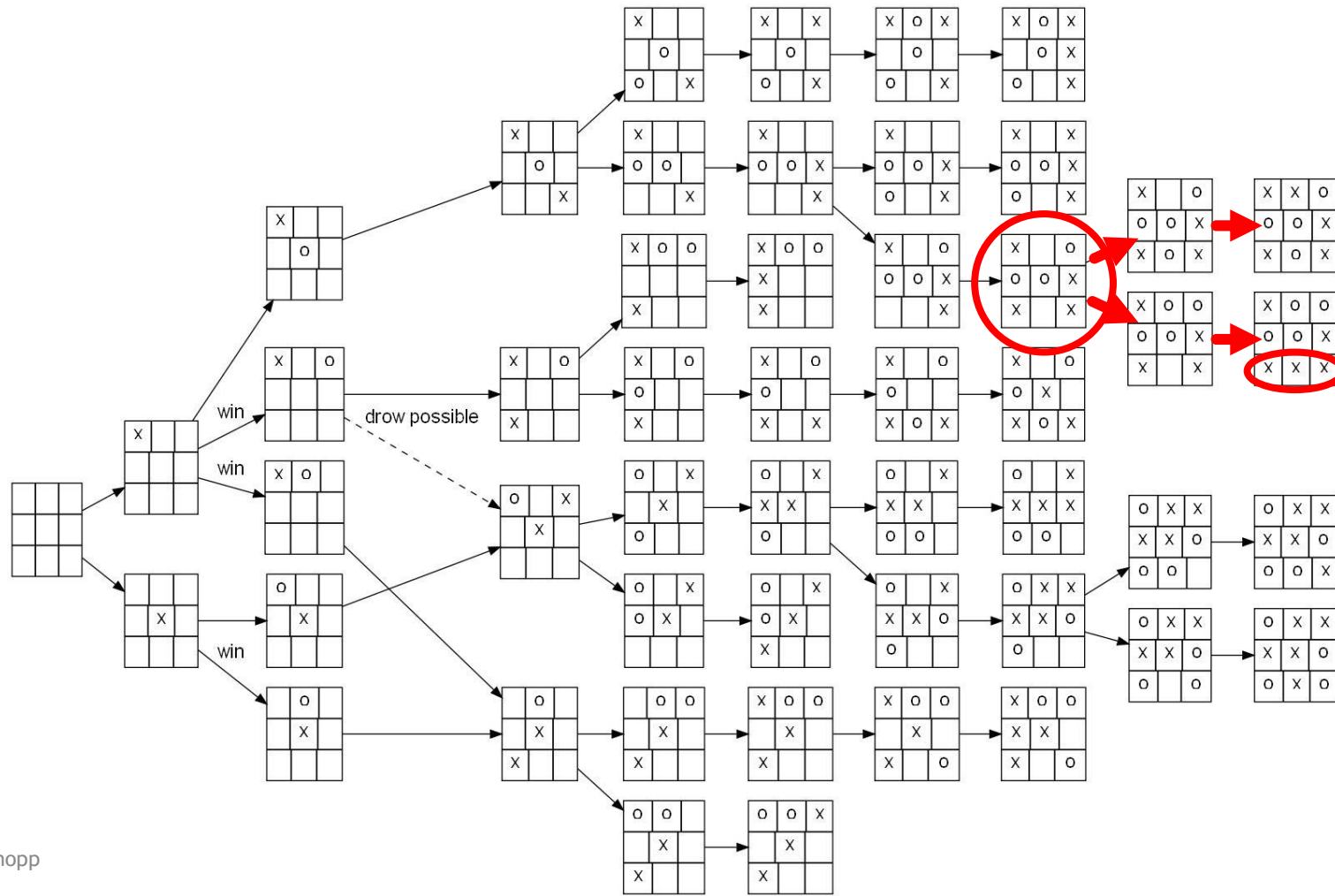
Traditioneller Ansatz für Brettspiele



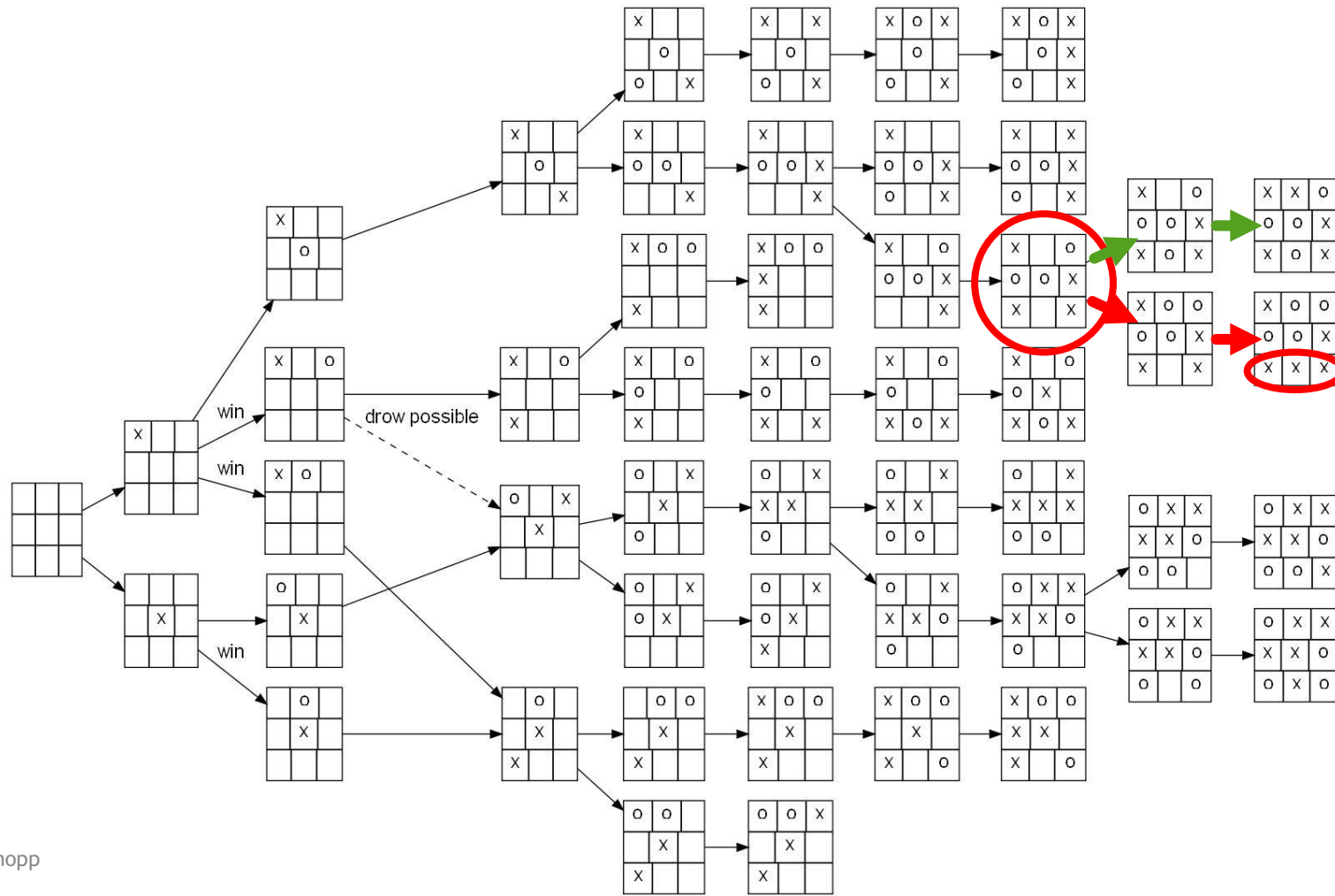
Traditioneller Ansatz für Brettspiele



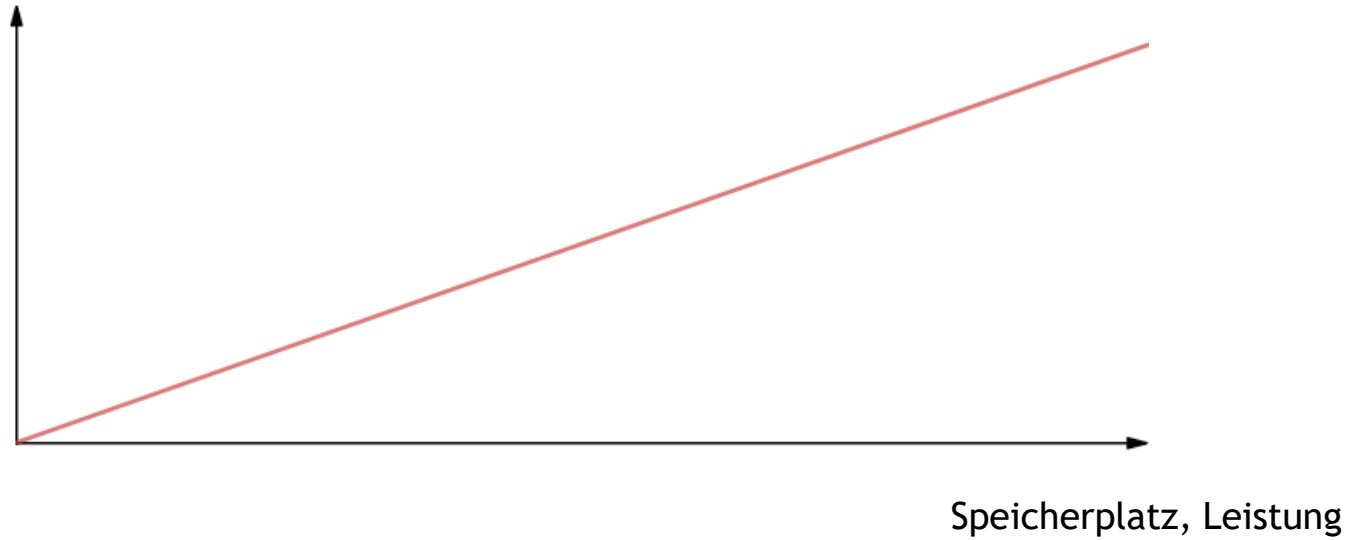
Traditioneller Ansatz für Brettspiele



Traditioneller Ansatz für Brettspiele



Komplexität Spiel



- ▶ Benötigte Leistung und Speicherplatz skaliert mit Komplexität des Spiels
- ▶ Nicht anwendbar auf komplexe Spiele mit sehr vielen Brettmöglichkeiten

Wie spielte Deep Blue Schach?

- ▶ Berechnung von 17mio möglichen Positionen
- ▶ Erzeuge mögliche Züge anhand der Regeln von Schach
- ▶ Evaluiere jeden möglichen Zug anhand der Evaluate-Funktion
- ▶ Wähle den bestmöglichen Zug aus

Wie spielte Deep Blue Schach?

- ▶ Game Tree zu groß um alle Pfade zu durchsuchen
- ▶ => Durchsuche so weit wie möglich (im Durchschnitt 6 Züge)
- ▶ Ersetze den SubTree durch die Evaluate-Funktion
 - ▶ Material
 - ▶ Position
 - ▶ King safety
 - ▶ Tempo
 - ▶ Expert domain knowledge
- ▶ Wähle den Zug mit dem höchsten Wert aus

Evaluate-Funktion

- ▶ Bestehend aus ca. 8.000 Teilen
- ▶ Viele extra auf bestimmte Positionen abgestimmt
- ▶ Opening Book:
 - ▶ 4.000 Positionen
 - ▶ 700.000 Grandmaster Spiele
- ▶ Endgame Database:
 - ▶ Viele Six-Piece endgames
 - ▶ Viele ≤ 5 Piecepositions

Mehr Brute Force als smarte AI

- ▶ Testen von sehr vielen möglichen Zügen
- ▶ Speziell entwickelte Evaluate-Funktionen mit menschlichem Expertenwissen
 - ▶ Overfitting
 - ▶ Maximal so gut wie die Entwickler der Funktion
- ▶ Nicht generalisierbar

Schach versus Go

Schach

- ▶ Beginn:
 - ▶ 16 Figuren, 6 verschiedenen Typen
 - ▶ Jeder Typ darf anders ziehen
- ▶ Ziel:
 - ▶ Gegnerischen König schlagen

Go

- ▶ Beginn:
 - ▶ Leeres Brett
 - ▶ Nur ein Figurentyp (Stein)
- ▶ Ziel:
 - ▶ Soviel Gebiet wie möglich einnehmen

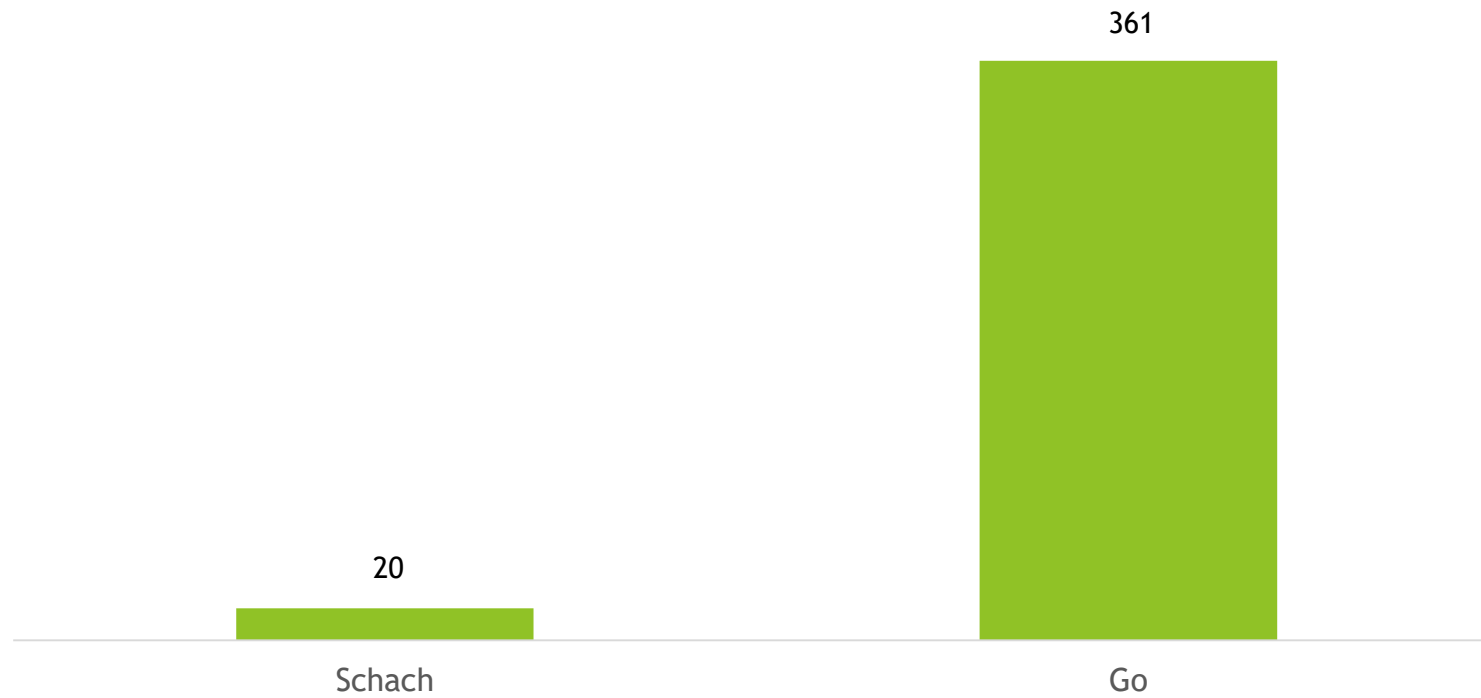
Schach versus Go

▶ Go besitzt einfachere Regeln

➔ Weniger Einschränkungen

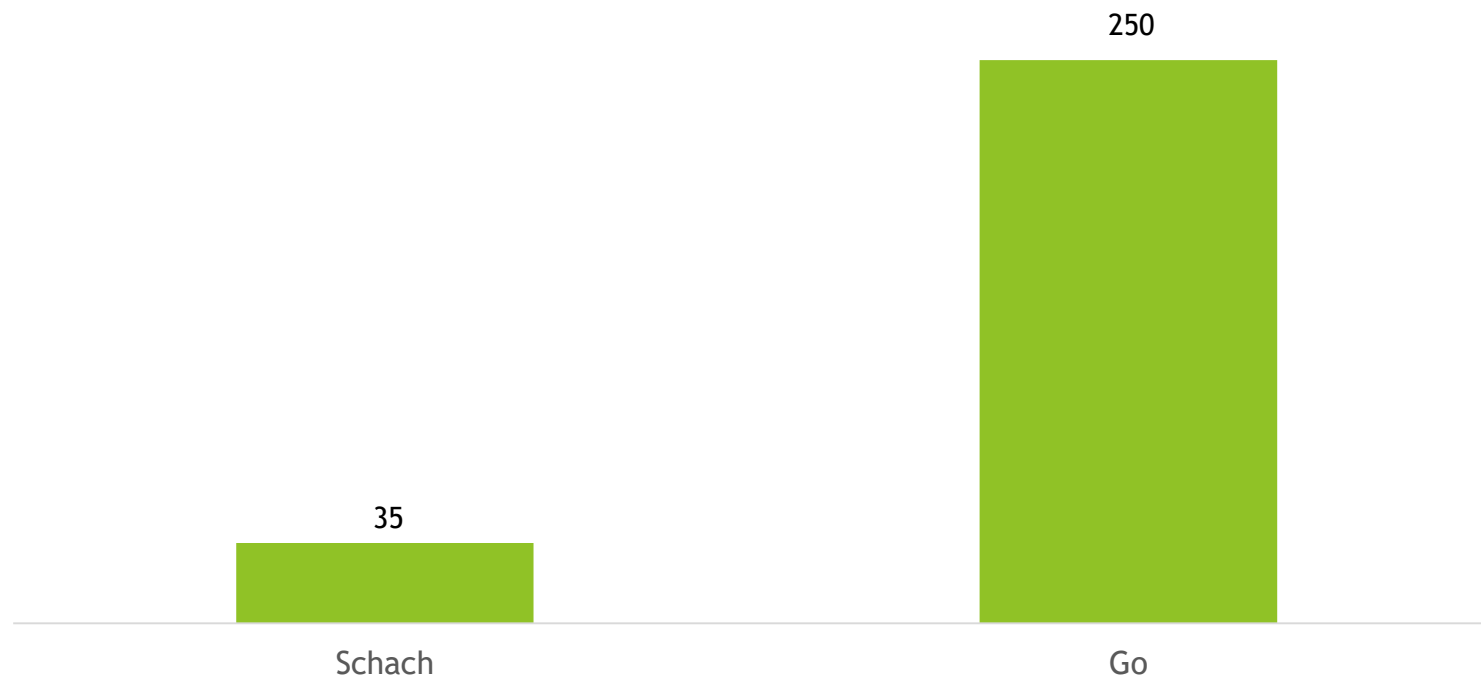
Schach versus Go

Eröffnung: mögliche Züge



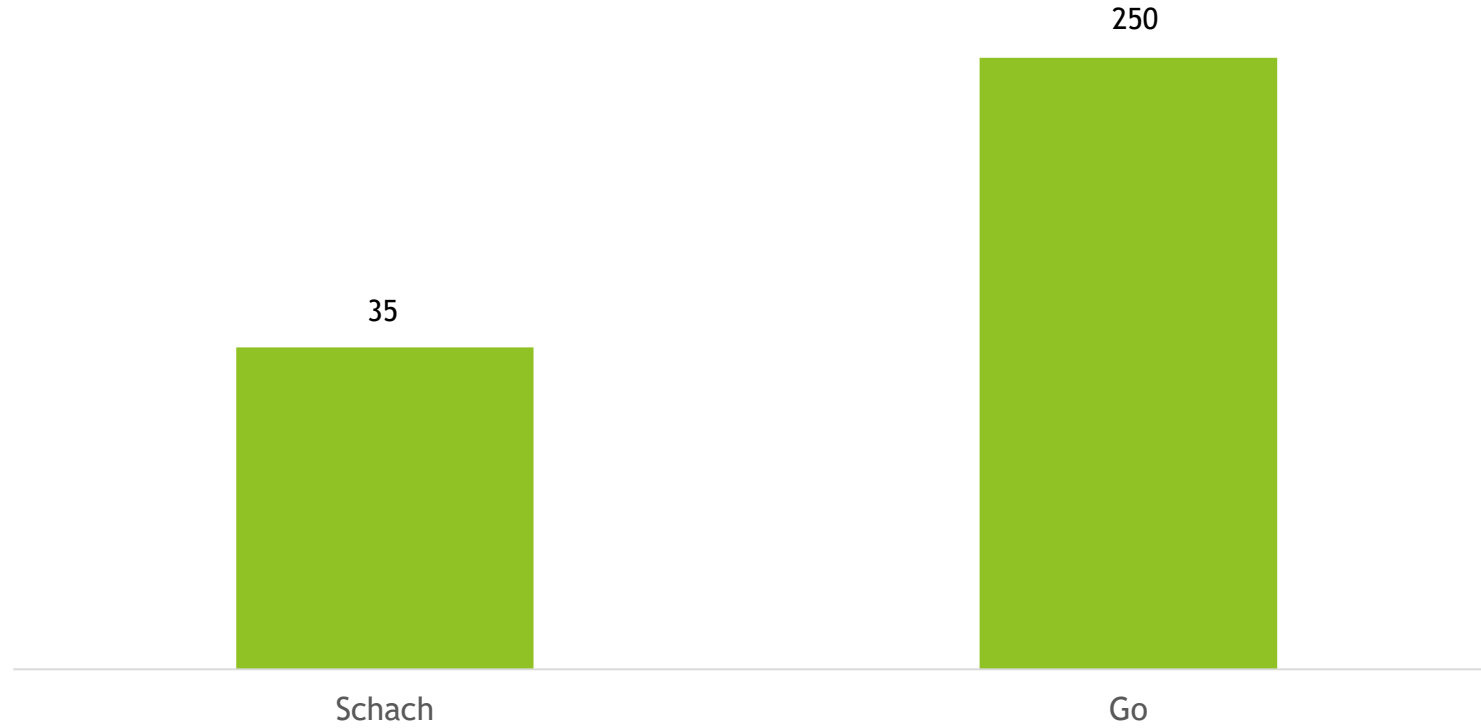
Schach versus Go

∅ mögliche Züge



Schach versus Go

Ø Spiellänge



Mögliche Boards Schach

10^{120}



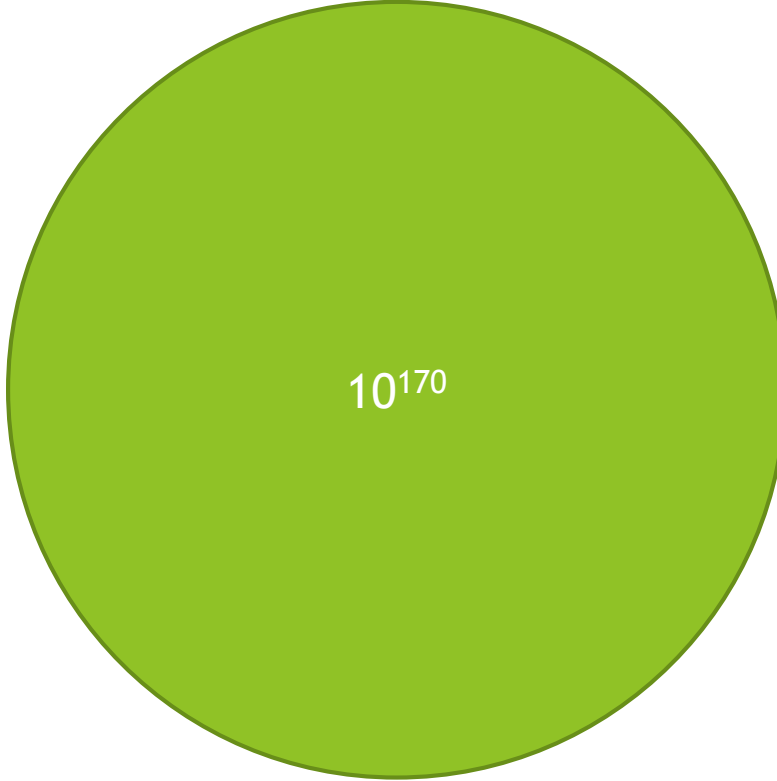
Mögliche Boards Schach

10^{120}



Mögliche Boards Go

10^{170}



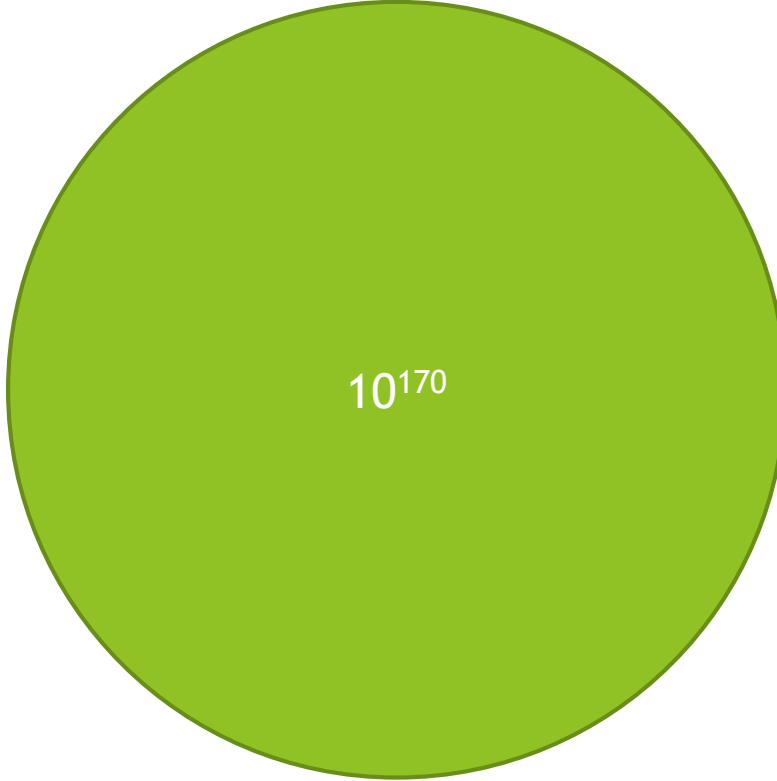
Mögliche Boards Schach

10^{120}



Mögliche Boards Go

10^{170}



Atome im Universum

10^{80}





Go ist zu komplex, um durch Verfahren wie IBM`s Deep Blue gelöst zu werden, geschweige denn vom traditionellen Ansatz

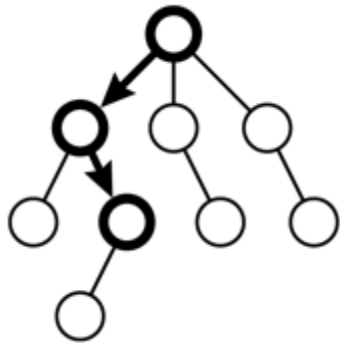
Wie konnte AlphaGo Lee Sedol schlagen?

Aufbau AlphaGo

AlphaGo



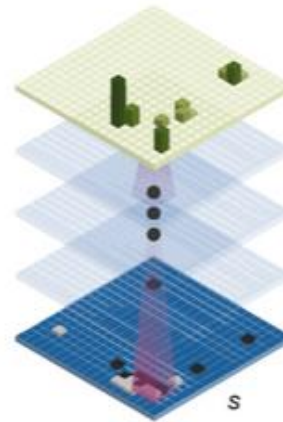
Monte Carlo Tree Search



[4] <http://ccg.doc.gold.ac.uk/research-mcts/>



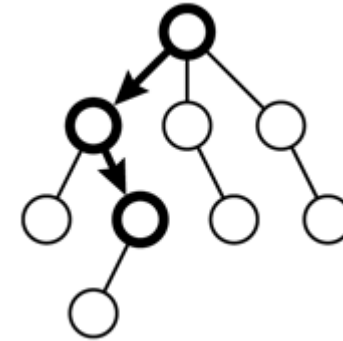
Deep Neural Networks



[5] <https://www.tastehit.com/blog/google-deepmind-alphago-how-it-works/>

Monte Carlo Tree Search

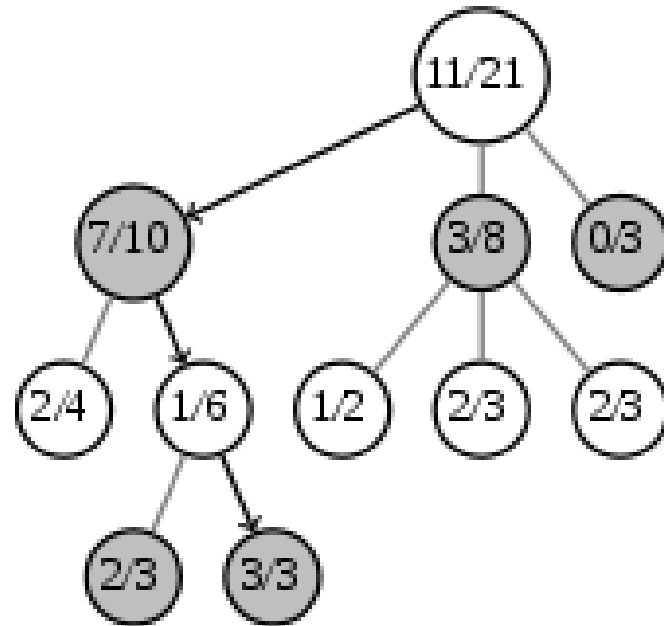
- ▶ Alternativer Ansatz Game Trees zu durchsuchen
- ▶ Simulation von möglichen Spielen
- ▶ Beginnt mit zufälligen Zügen
- ▶ Je mehr Simulationen, desto besser der Zug
 - ▶ MCTS konvergiert zum bestmöglichen Zug



[4] <http://ccg.doc.gold.ac.uk/research-mcts/>

Monte Carlo Tree Search

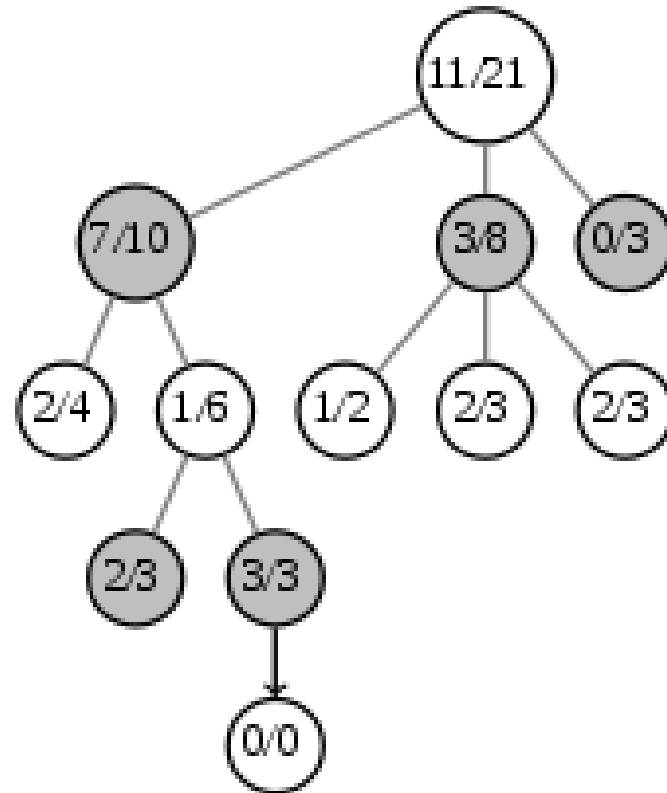
Selection:



[7] https://en.wikipedia.org/wiki/Monte_Carlo_tree_search

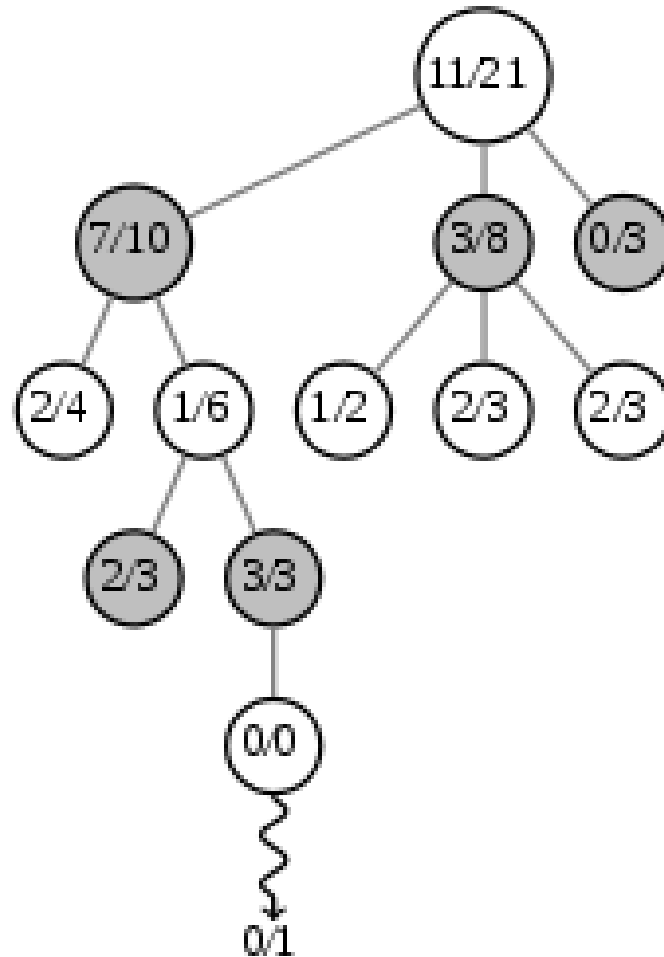
Monte Carlo Tree Search

Expansion:



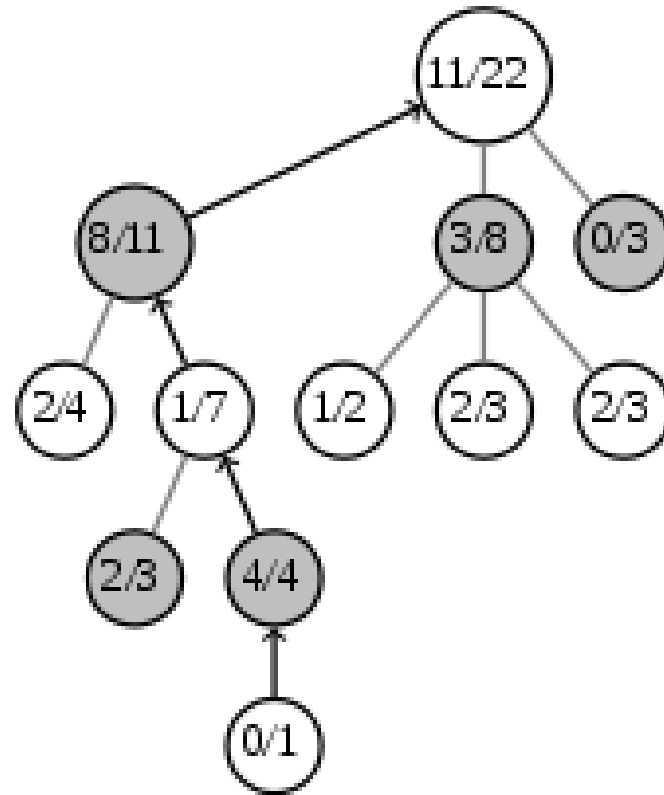
Monte Carlo Tree Search

Simulation:



Monte Carlo Tree Search

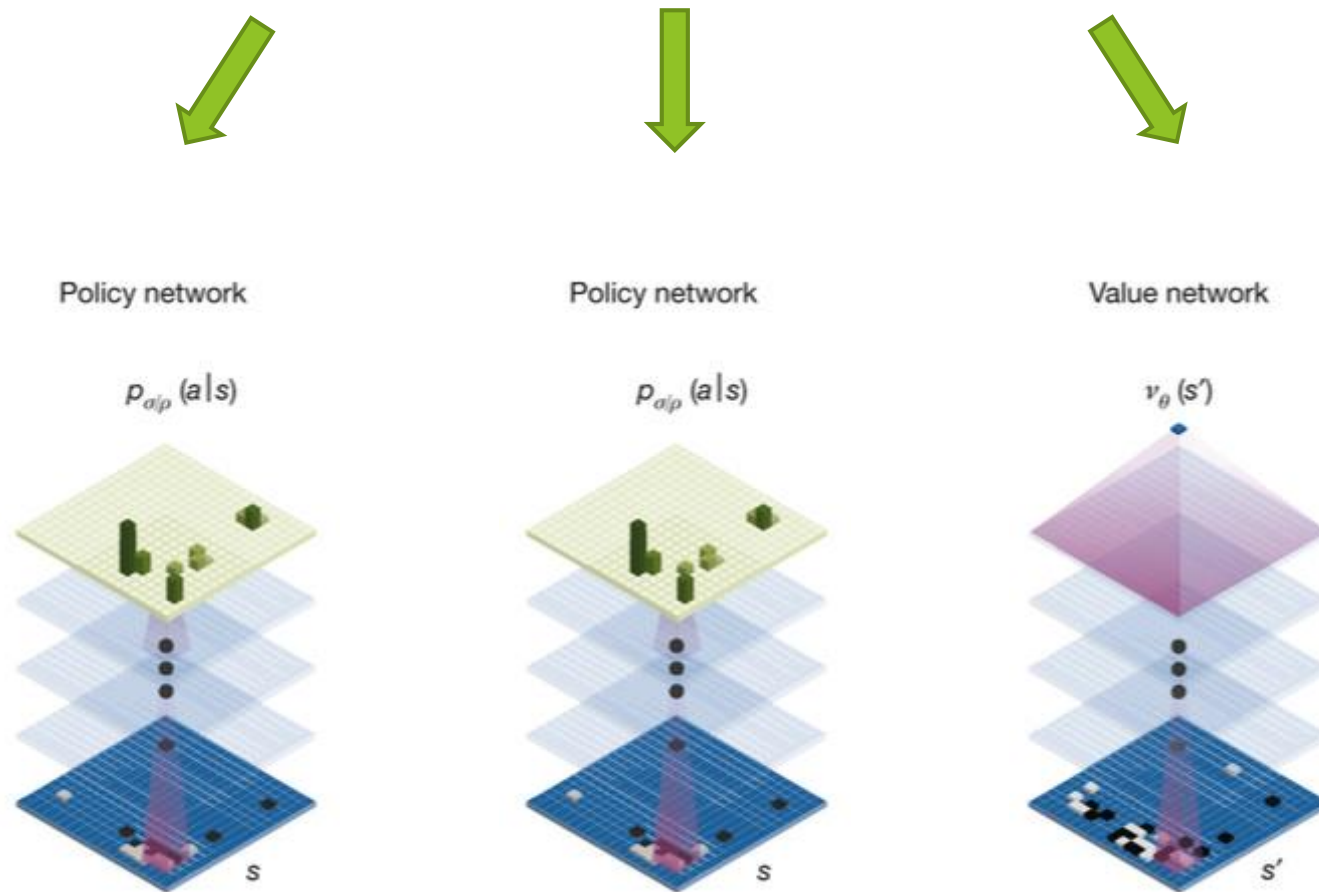
Backpropagation:



Effektivität von MCTS

- ▶ Reine Berechnung durch MCTS
 - ▶ MCTS kombiniert mit menschlichem Fachwissen
- ➔ Niveau auf Level von starkem Amateur-Spiel

Deep Neural Networks



Policy Network

- ▶ Input: Aktuelles Board als Bild
- ▶ Leitet die MCTS
- ▶ Output: Wertigkeit eines jeden möglichen Zugs
- ▶ Auswahl der bestmöglichen Züge
- ▶ Reduzierung der Breite des Suchbaums

Heavy vs. Fast Rollout

HeavyRollout

- ▶ 57% Genauigkeit
- ▶ 3ms Zugzeit

Fast Rollout

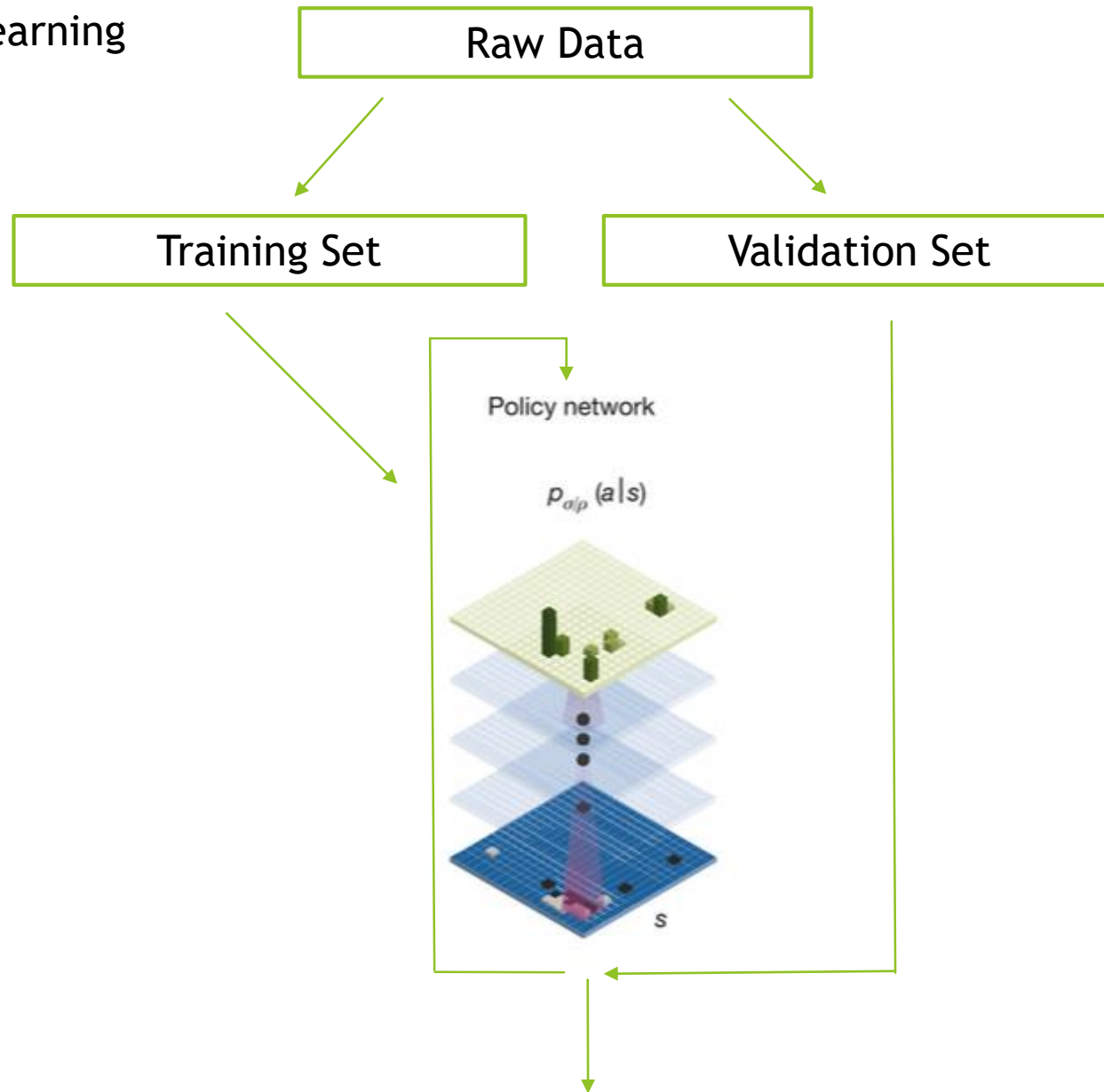
- ▶ Weniger Neuronen / Synapsen
- ▶ 24% Genauigkeit
- ▶ 2 μ s Zugzeit

Policy Network: Training

Stage 1: Supervised Learning

- ▶ Trainiert mit 30 Millionen Positionen von Spielen von menschlichen Spielern
 - ▶ KGS Go Server
- ▶ 57% Genauigkeit den menschlichen Zug vorauszusagen
- ▶ Ø Zugzeit: 3ms

Supervised Learning



Policy Network: Training

Stage 2: Reinforcement Learning

- ▶ Spielt gegen zufällige vorherige Iterationen von sich selbst
 - ▶ Minimierung von Overfitting
- ▶ Ausgang des Spiels wird als Trainingssignal verwendet
- ▶ 80% Winrate gegen das SL Policy Network
- ▶ 85% Winrate vs Pachi
 - ▶ Pachi: Nur MCTS mit 100.000 Simulationen pro Zug

Value Network

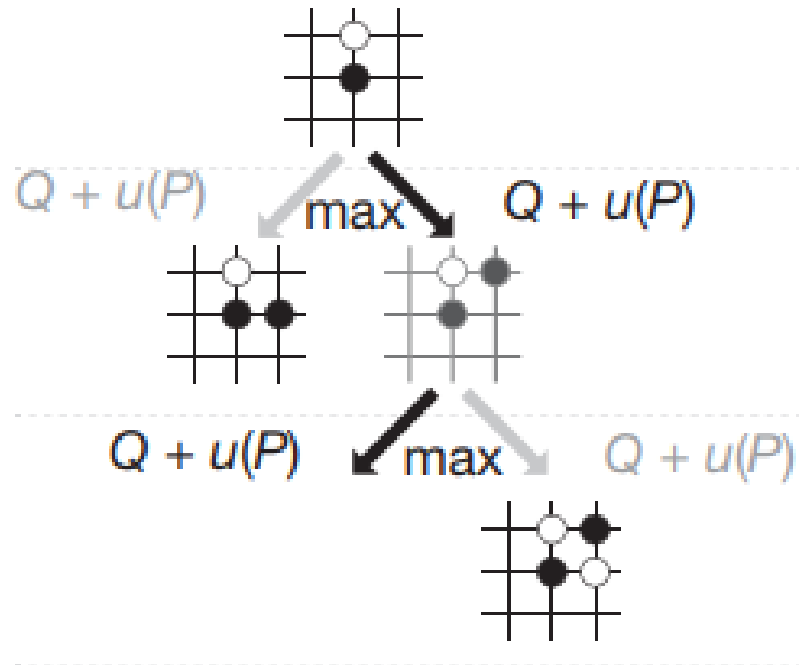
- ▶ Input: Aktuelles Board als Bild
- ▶ Wertet den aktuellen Stand des Spiels
 - ▶ Wie hoch ist die Wahrscheinlichkeit, dass der schwarze Spieler gewinnt?
- ▶ Output: Einzelne Nummer, Gewinnchance von Schwarz
- ▶ Ersetzt die Evaluate-Funktion von Deep Blue
 - ▶ Gelernt, nicht definiert
 - ▶ Kein menschliches Wissen notwendig
 - ▶ Keine Limitierung durch den Menschen
- ▶ Reduziert die Tiefe des Suchbaums

Value Network: Training

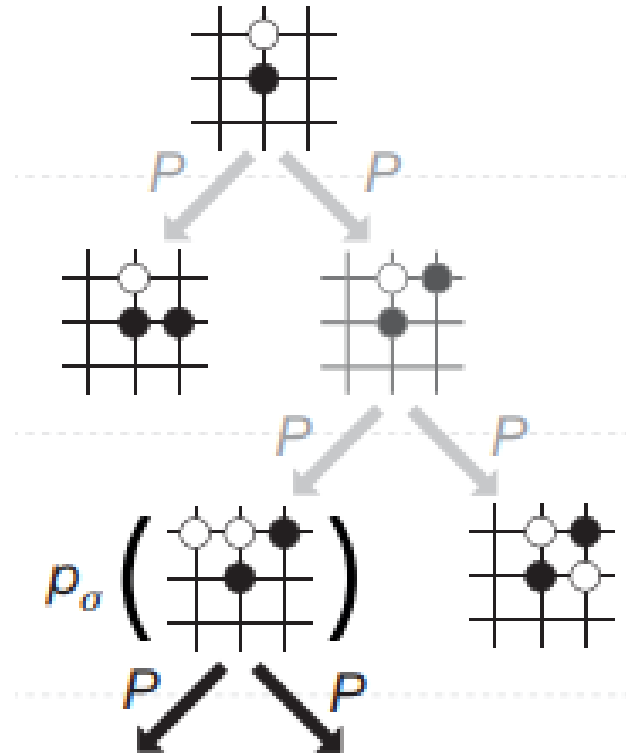
- ▶ Training mit KGS Go Daten führt zu Overfitting
- ▶ Trainiert mit 30 Millionen Positionen von generierten Self-Play data
- ▶ Jede Position aus einem anderen Spiel zwischen dem RL Policy Network und sich selbst
- ▶ Genauer als MCTS mit fast rollout policy network
- ▶ Ähnlich genau wie MCTS mit heavy policy network
- ▶ 15.000 mal weniger Rechenleistung

Kombination MCTS und Neural Networks

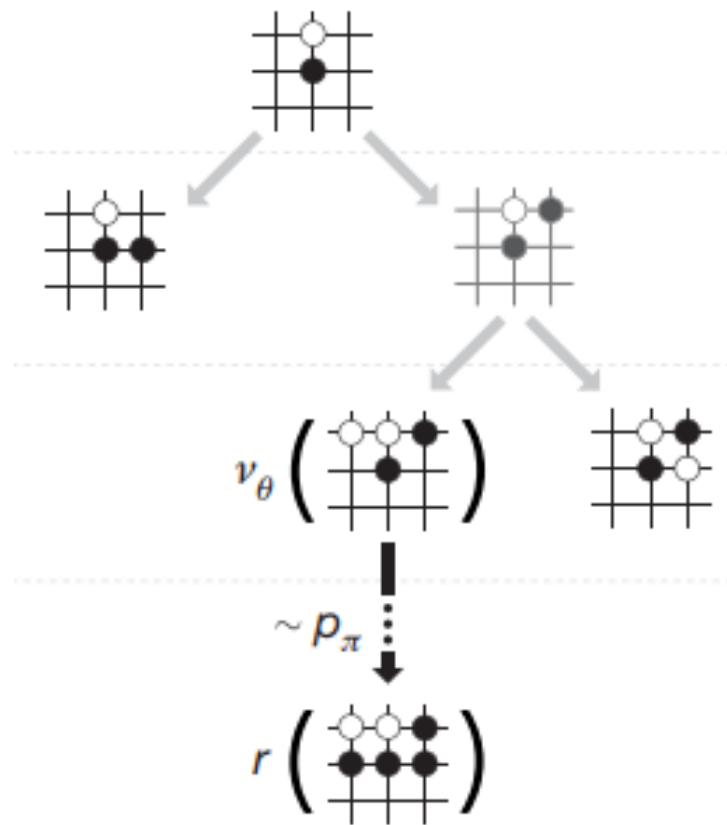
Selection



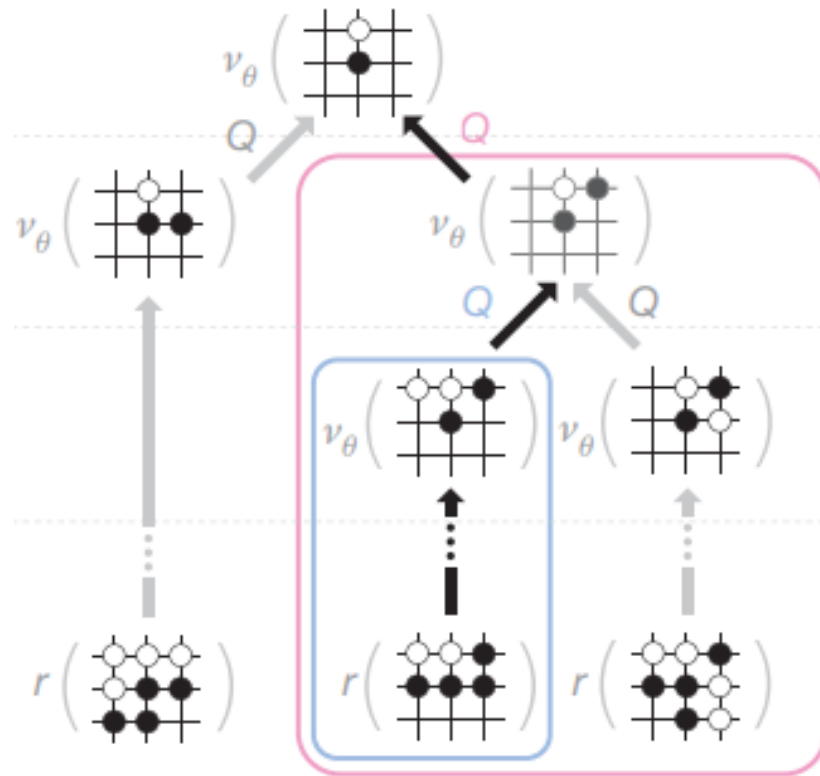
Expansion

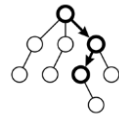
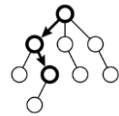
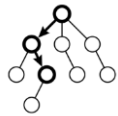
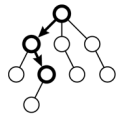


Evaluation

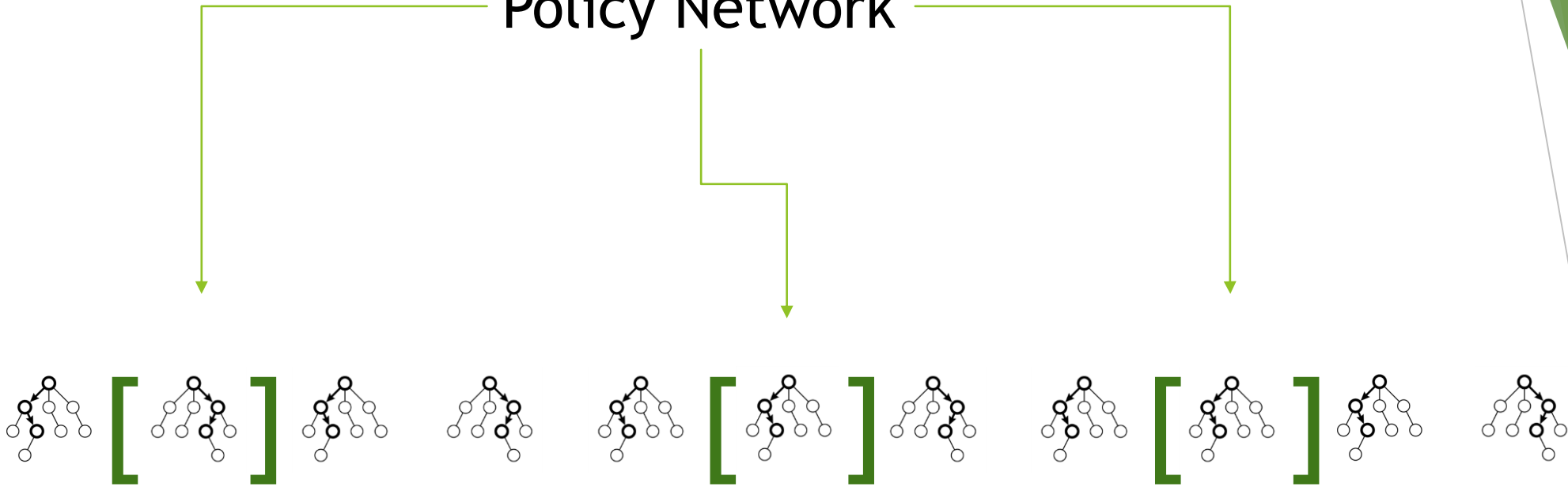


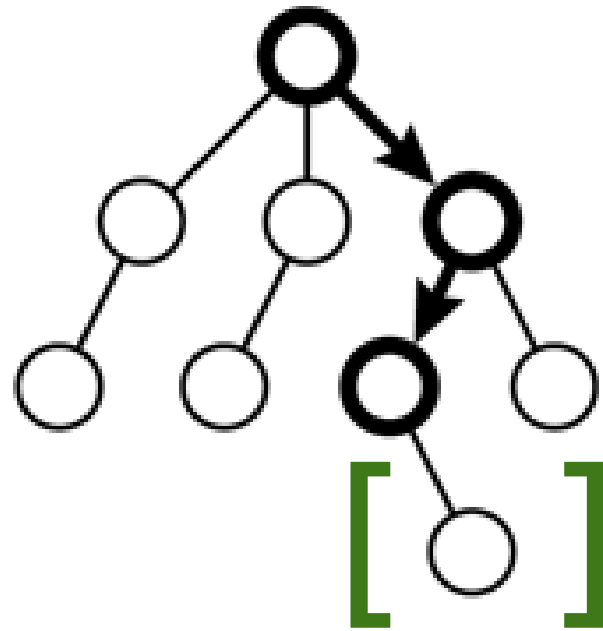
Back-Propagation





Policy Network





Value Network

[4] <http://ccg.doc.gold.ac.uk/research-mcts/>

Wie kann das neue Wissen genutzt werden?

- ▶ Anwendung in der Medizinischen Forschung
 - ▶ Falten von Proteinen
 - ▶ IBM's Watson for Oncology
- ▶ AlphaGo Zero
 - ▶ „Tabula Rasa“-Learning
 - ▶ General Purpose AI

Quellen

- ▶ [4] <http://ccg.doc.gold.ac.uk/research-mcts/>
- ▶ [5] <https://www.tastehit.com/blog/google-deepmind-alphago-how-it-works/>
- ▶ <https://jeffbradberry.com/posts/2015/09/intro-to-monte-carlo-tree-search/>
- ▶ <https://www.tastehit.com/blog/google-deepmind-alphago-how-it-works/>
- ▶ <http://ccg.doc.gold.ac.uk/research-mcts/>
- ▶ <https://machinelearnings.co/understanding-alphago-948607845bb1>
- ▶ <https://deepmind.com/blog/alphago-zero-learning-scratch/>
- ▶ <https://deepmind.com/research/alphago/>
- ▶ <https://blog.google/topics/machine-learning/alphago-machine-learning-game-go/>
- ▶ [6] Mastering the Game of Go with Deep Neural Networks and Tree Search
 - ▶ Google, Google DeepMind | Published in Nature 529, Januar 2016

Quellen

- ▶ Mastering the game of Go without human knowledge
 - ▶ Google, Google DeepMind | Published in Nature 550, April 2017
- ▶ [7] https://en.wikipedia.org/wiki/Monte_Carlo_tree_search
- ▶ [1] <http://www.businessinsider.com/how-ibm-watson-is-transforming-healthcare-2015-7?IR=T>
- ▶ [2] <https://sciencebasedmedicine.org/tag/watson/>
- ▶ [3] <https://www.newscientist.com/article/2079871-im-in-shock-how-an-ai-beat-the-worlds-best-human-at-go/>
- ▶ <http://www.businessinsider.com/r-ibms-watson-to-guide-cancer-therapies-at-14-centers-2015-5?IR=T>

Quellen

- ▶ <https://www.engadget.com/2017/06/01/ibm-watson-cancer-treatment-plans/>
- ▶ <http://www.telegraph.co.uk/science/2017/10/18/alphago-zero-google-deepmind-supercomputer-learns-3000-years/>
- ▶ <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4828734/>
- ▶ https://en.wikipedia.org/wiki/Deep_Blue_versus_Garry_Kasparov
- ▶ [https://en.wikipedia.org/wiki/Watson_\(computer\)](https://en.wikipedia.org/wiki/Watson_(computer))