AlphaZero

Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm

Karel Ha
article by Google DeepMind

AI Seminar, 19th December 2017

Outline

The Alpha* Timeline

Fan Hui

https://en.wikipedia.org/wiki/Fan_Hui

Fan Hui

- professional 2 dan

Fan Hui

- professional 2 dan
- European Go Champion in 2013, 2014 and 2015

https://en.wikipedia.org/wiki/Fan_Hui

Fan Hui

- professional 2 dan
- European Go Champion in 2013, 2014 and 2015
- European Professional Go Champion in 2016

https://en.wikipedia.org/wiki/Fan_Hui

AlphaGo (AlphaGo Fan) vs. Fan Hui

AlphaGo (AlphaGo Fan) vs. Fan Hui

AlphaGo won 5:0 in a formal match on October 2015.

AlphaGo (AlphaGo Fan) vs. Fan Hui

AlphaGo won 5:0 in a formal match on October 2015.

> [AlphaGo] is very strong and stable, it seems like a wall. ... I know AlphaGo is a computer, but if no one told me, maybe I would think the player was a little strange, but a very strong player, a real person.
>
> — Fan Hui

Lee Sedol “The Strong Stone”

https://en.wikipedia.org/wiki/Lee_Sedol

Lee Sedol "The Strong Stone"

- professional 9 dan

https://en.wikipedia.org/wiki/Lee_Sedol

Lee Sedol "The Strong Stone"

- professional 9 dan
- the 2^{nd} in international titles

Lee Sedol “The Strong Stone”

- professional 9 dan
- the 2^{nd} in international titles
- the 5^{th} youngest to become a professional Go player in South Korean history

https://en.wikipedia.org/wiki/Lee_Sedol

Lee Sedol “The Strong Stone”

- professional 9 dan
- the 2^{nd} in international titles
- the 5^{th} youngest to become a professional Go player in South Korean history
- Lee Sedol would win 97 out of 100 games against Fan Hui.

https://en.wikipedia.org/wiki/Lee_Sedol

Lee Sedol "The Strong Stone"

- professional 9 dan
- the 2^{nd} in international titles
- the 5^{th} youngest to become a professional Go player in South Korean history
- Lee Sedol would win 97 out of 100 games against Fan Hui.
- "Roger Federer" of Go

https://en.wikipedia.org/wiki/Lee_Sedol

I heard Google DeepMind's AI is surprisingly strong and getting stronger, but I am confident that I can win, at least this time.

Lee Sedol


I heard Google DeepMind's AI is surprisingly strong and getting stronger, but I am confident that I can win, at least this time.
Lee Sedol
...even beating AlphaGo by 4:1 may allow the Google DeepMind team to claim its de facto victory and the defeat of him [Lee Sedol], or even humankind.
interview in JTBC Newsroom

I heard Google DeepMind's AI is surprisingly strong and getting stronger, but I am confident that I can win, at least this time.

Lee Sedol

...even beating AlphaGo by 4:1 may allow the Google DeepMind team to claim its de facto victory and the defeat of him [Lee Sedol], or even humankind.

interview in JTBC Newsroom

AlphaGo (AlphaGo Lee) vs. Lee Sedol

https://en.wikipedia.org/wiki/AlphaGo_versus_Lee_Sedol

AlphaGo (AlphaGo Lee) vs. Lee Sedol

In March 2016 **AlphaGo won 4:1** against the legendary Lee Sedol.

https://en.wikipedia.org/wiki/AlphaGo_versus_Lee_Sedol

AlphaGo (AlphaGo Lee) vs. Lee Sedol

In March 2016 **AlphaGo won 4:1** against the legendary Lee Sedol.

AlphaGo won all but the 4^{th} game; all games were won by resignation.

https://en.wikipedia.org/wiki/AlphaGo_versus_Lee_Sedol

AlphaGo (AlphaGo Lee) vs. Lee Sedol

In March 2016 **AlphaGo won 4:1** against the legendary Lee Sedol.

AlphaGo won all but the 4^{th} game; all games were won by resignation.

https://en.wikipedia.org/wiki/AlphaGo_versus_Lee_Sedol

AlphaGo Master

https://deepmind.com/research/alphago/match-archive/master/

AlphaGo Master

In January 2017, DeepMind revealed that AlphaGo had played a series of unofficial online games against some of the strongest professional Go players under the pseudonyms “Master” and ”Magister”.

https://deepmind.com/research/alphago/match-archive/master/

AlphaGo Master

In January 2017, DeepMind revealed that AlphaGo had played a series of unofficial online games against some of the strongest professional Go players under the pseudonyms “Master” and ”Magister”.

This AlphaGo was an improved version of the AlphaGo that played Lee Sedol in 2016.

https://deepmind.com/research/alphago/match-archive/master/

AlphaGo Master

In January 2017, DeepMind revealed that AlphaGo had played a series of unofficial online games against some of the strongest professional Go players under the pseudonyms “Master” and ”Magister”.

This AlphaGo was an improved version of the AlphaGo that played Lee Sedol in 2016.

Over one week, AlphaGo played 60 online fast time-control games.

https://deepmind.com/research/alphago/match-archive/master/

AlphaGo Master

In January 2017, DeepMind revealed that AlphaGo had played a series of unofficial online games against some of the strongest professional Go players under the pseudonyms “Master” and ”Magister”.

This AlphaGo was an improved version of the AlphaGo that played Lee Sedol in 2016.

Over one week, AlphaGo played 60 online fast time-control games.

AlphaGo won this series of games 60:0.

https://deepmind.com/research/alphago/match-archive/master/


中国围棋协会
Google
浙江省体育局
中国乌镇 围棋峰会 顶尖棋手 + DeepMind AlphaGo 共创棋妙未来
The Future of Go Summit in Wuzhen
Legendary players and DeepMind's AlphaGo explore the mysteries of Go

https://events.google.com/alphago2017/

- 23 May - 27 May 2017 in Wuzhen, China

https://events.google.com/alphago2017/

- 23 May - 27 May 2017 in Wuzhen, China
- Team Go vs. AlphaGo **0:1**

https://events.google.com/alphago2017/

- 23 May - 27 May 2017 in Wuzhen, China
- Team Go vs. AlphaGo **0:1**
- AlphaGo vs. world champion Ke Jie **3:0**

https://events.google.com/alphago2017/


AlphaGo Zero
Starting from scratch

defeated AlphaGo Lee by 100 games to 0

AI system that mastered chess, Shogi and Go to "superhuman levels" within a handful of hours

AlphaZero

AI system that mastered chess, Shogi and Go to "superhuman levels" within a handful of hours

AlphaZero

defeated AlphaGo Zero (version with 20 blocks trained for 3 days) by **60 games to 40**

1 **AlphaGo Fan**

1. AlphaGo Fan
2. AlphaGo Lee

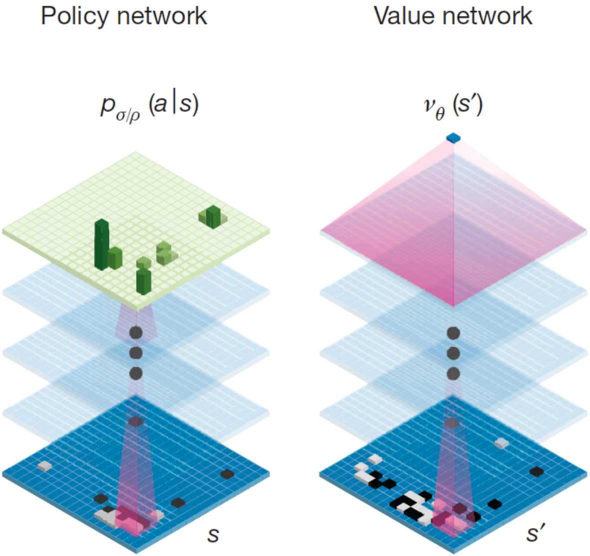
1. AlphaGo Fan
2. AlphaGo Lee
3. **AlphaGo Master**

1. AlphaGo Fan
2. AlphaGo Lee
3. AlphaGo Master
4. **AlphaGo Zero**

1. **AlphaGo Fan**
2. **AlphaGo Lee**
3. **AlphaGo Master**
4. **AlphaGo Zero**
5. **AlphaZero**

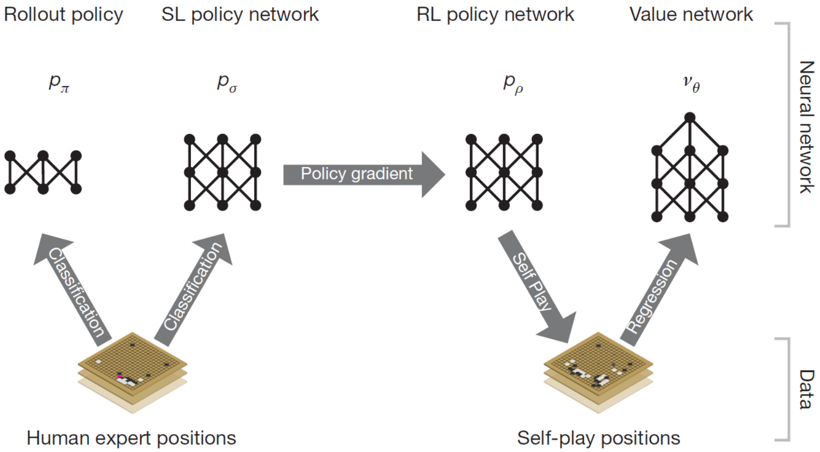
AlphaGo

Policy and Value Networks



[Silver et al. 2016]

Training the (Deep Convolutional) Neural Networks



[Silver et al. 2016]

AlphaGo Zero (AG0)

AG0: Differences Compared to AlphaGo {Fan, Lee, Master}

AlphaGo {Fan, Lee, Master} $\times$ AlphaGo Zero:

[Silver et al. 2017b]

AG0: Differences Compared to AlphaGo {Fan, Lee, Master}

AlphaGo {Fan, Lee, Master} $\times$ AlphaGo Zero:

- supervised learning from human expert positions $\times$ from scratch by self-play reinforcement learning ("tabula rasa")

[Silver et al. 2017b]

AG0: Differences Compared to AlphaGo {Fan, Lee, Master}

AlphaGo {Fan, Lee, Master} $\times$ AlphaGo Zero:

- supervised learning from human expert positions $\times$ from scratch by self-play reinforcement learning ("tabula rasa")
- additional (auxialiary) input features $\times$ only the black and white stones from the board as input features

[Silver et al. 2017b]

AG0: Differences Compared to AlphaGo {Fan, Lee, Master}

AlphaGo {Fan, Lee, Master} × AlphaGo Zero:

- supervised learning from human expert positions × from scratch by self-play reinforcement learning ("tabula rasa")
- additional (auxialiary) input features × only the black and white stones from the board as input features
- separate policy and value networks × single neural network

[Silver et al. 2017b]

AG0: Differences Compared to AlphaGo {Fan, Lee, Master}

AlphaGo {Fan, Lee, Master} × AlphaGo Zero:

- supervised learning from human expert positions × from scratch by self-play reinforcement learning ("tabula rasa")
- additional (auxialiary) input features × only the black and white stones from the board as input features
- separate policy and value networks × single neural network
- tree search using also Monte Carlo rollouts × simpler tree search using only the single neural network to both evaluate positions and sample moves

[Silver et al. 2017b]

AG0: Differences Compared to AlphaGo {Fan, Lee, Master}

AlphaGo {Fan, Lee, Master} × AlphaGo Zero:

- supervised learning from human expert positions × from scratch by self-play reinforcement learning ("tabula rasa")
- additional (auxialiary) input features × only the black and white stones from the board as input features
- separate policy and value networks × single neural network
- tree search using also Monte Carlo rollouts × simpler tree search using only the single neural network to both evaluate positions and sample moves
- (AlphaGo Lee) distributed machines + 48 tensor processing units (TPUs) × single machines + 4 TPUs

[Silver et al. 2017b]

AG0: Differences Compared to AlphaGo {Fan, Lee, Master}

AlphaGo {Fan, Lee, Master} × AlphaGo Zero:

- supervised learning from human expert positions × from scratch by self-play reinforcement learning ("tabula rasa")
- additional (auxialiary) input features × only the black and white stones from the board as input features
- separate policy and value networks × single neural network
- tree search using also Monte Carlo rollouts × simpler tree search using only the single neural network to both evaluate positions and sample moves
- (AlphaGo Lee) distributed machines + 48 tensor processing units (TPUs) × single machines + 4 TPUs
- (AlphaGo Lee) several months of training time × 72 h of training time (outperforming AlphaGo Lee after 36 h)

[Silver et al. 2017b]

AG0 achieves this via

AG0 achieves this via

- a new reinforcement learning algorithm

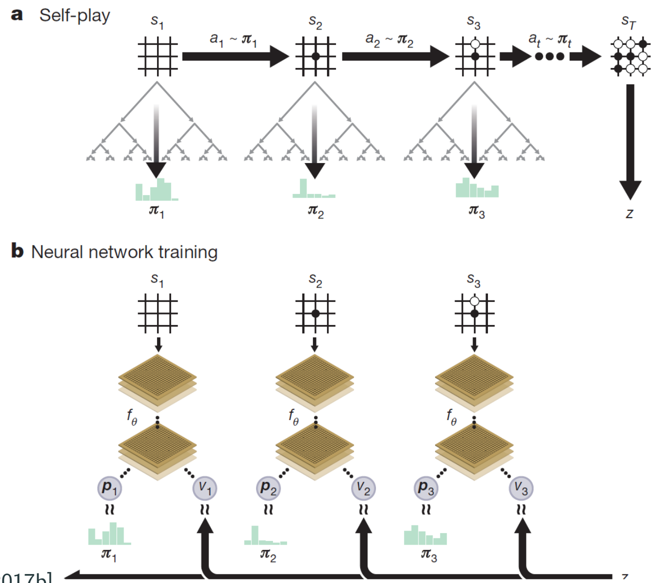
AG0 achieves this via

- a new reinforcement learning algorithm
- with lookahead search inside the training loop

[Silver et al. 2017b]

AG0: Self-Play Reinforcement Learning

AG0: Self-Play Reinforcement Learning



[Silver et al. 2017b]

AG0: Self-Play Reinforcement Learning – Neural Network

deep neural network f_θ with parameters θ:

AG0: Self-Play Reinforcement Learning – Neural Network

deep neural network f_θ with parameters θ:

- input: raw board representation s

[Silver et al. 2017b]

AG0: Self-Play Reinforcement Learning – Neural Network

deep neural network f_θ with parameters θ:

- input: raw board representation s
- output:

AG0: Self-Play Reinforcement Learning – Neural Network

deep neural network f_θ with parameters θ:

- input: raw board representation s
- output:
  - move probabilities $\mathbf{p}$

[Silver et al. 2017b]

AG0: Self-Play Reinforcement Learning – Neural Network

deep neural network f_θ with parameters θ:

- input: raw board representation s
- output:
  - move probabilities $\mathbf{p}$
  - value v of the board position

[Silver et al. 2017b]

AG0: Self-Play Reinforcement Learning – Neural Network

deep neural network f_θ with parameters θ:

- input: raw board representation s
- output:
  - move probabilities $\mathbf{p}$
  - value v of the board position
  - $f_\theta(s) = (\mathbf{p}, v)$

[Silver et al. 2017b]

AG0: Self-Play Reinforcement Learning – Neural Network

deep neural network f_θ with parameters θ:

- input: raw board representation s
- output:
  - move probabilities $\mathbf{p}$
  - value v of the board position
  - $f_\theta(s) = (\mathbf{p}, v)$
- specifics:

[Silver et al. 2017b]

AG0: Self-Play Reinforcement Learning – Neural Network

deep neural network f_θ with parameters θ:

- input: raw board representation s
- output:
  - move probabilities $\mathbf{p}$
  - value v of the board position
  - $f_\theta(s) = (\mathbf{p}, v)$
- specifics:
  - (20 or 40) residual blocks (of convolutional layers)

[Silver et al. 2017b]

AG0: Self-Play Reinforcement Learning – Neural Network

deep neural network f_θ with parameters θ:

- input: raw board representation s
- output:
  - move probabilities $\mathbf{p}$
  - value v of the board position
  - $f_\theta(s) = (\mathbf{p}, v)$
- specifics:
  - (20 or 40) residual blocks (of convolutional layers)
  - batch normalization

[Silver et al. 2017b]

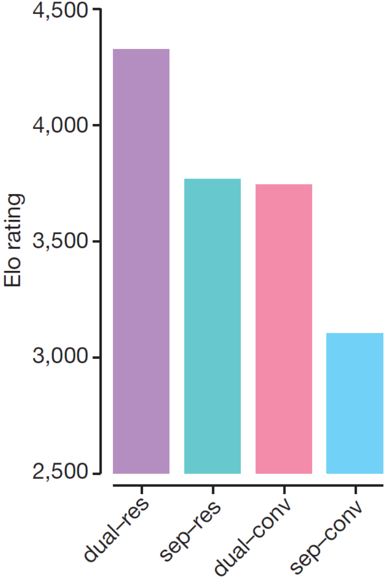
AG0: Self-Play Reinforcement Learning – Neural Network

deep neural network f_θ with parameters θ:

- input: raw board representation s
- output:
  - move probabilities $\mathbf{p}$
  - value v of the board position
  - $f_\theta(s) = (\mathbf{p}, v)$
- specifics:
  - (20 or 40) residual blocks (of convolutional layers)
  - batch normalization
  - rectifier non-linearities

[Silver et al. 2017b]

AG0: Comparison of Various Neural Network Architectures



[Silver et al. 2017b]

AG0: Self-Play Reinforcement Learning – Steps

0. random weights θ_0

AG0: Self-Play Reinforcement Learning – Steps

0. random weights θ_0
1. at each iteration $i > 0$, self-play games are generated:

AG0: Self-Play Reinforcement Learning – Steps

0. random weights θ_0
1. at each iteration $i > 0$, self-play games are generated:
   i. MCTS samples search probabilities π_t based on the neural network from the previous iteration $f_{\theta_{i-1}}$:

   $$\pi_t = \alpha_{\theta_{i-1}}(s_t)$$

   for each time-step $t = 1, 2, \ldots, T$

[Silver et al. 2017b]

AG0: Self-Play Reinforcement Learning – Steps

0. random weights θ_0
1. at each iteration $i > 0$, self-play games are generated:
   i. MCTS samples search probabilities π_t based on the neural network from the previous iteration $f_{\theta_{i-1}}$:

   $$\pi_t = \alpha_{\theta_{i-1}}(s_t)$$

   for each time-step $t = 1, 2, \ldots, T$

   ii. move is sampled from π_t

[Silver et al. 2017b]

AG0: Self-Play Reinforcement Learning – Steps

0. random weights θ_0
1. at each iteration $i > 0$, self-play games are generated:
   i. MCTS samples search probabilities π_t based on the neural network from the previous iteration $f_{\theta_{i-1}}$:

   $$\pi_t = \alpha_{\theta_{i-1}}(s_t)$$

   for each time-step $t = 1, 2, \ldots, T$

   ii. move is sampled from π_t

   iii. data (s_t, π_t, z_t) for each t are stored for later training

[Silver et al. 2017b]

AG0: Self-Play Reinforcement Learning – Steps

0. random weights θ_0
1. at each iteration $i > 0$, self-play games are generated:
   i. MCTS samples search probabilities π_t based on the neural network from the previous iteration $f_{\theta_{i-1}}$:

   $$\pi_t = \alpha_{\theta_{i-1}}(s_t)$$

   for each time-step $t = 1, 2, \ldots, T$

   ii. move is sampled from π_t

   iii. data (s_t, π_t, z_t) for each t are stored for later training

   iv. new neural network f_{θ_i} is trained in order to minimize the loss

   $$l = (z - v)^2 - \pi^\top \log \mathbf{p} + c||\theta||^2$$

[Silver et al. 2017b]

AG0: Self-Play Reinforcement Learning – Steps

0. random weights θ_0
1. at each iteration $i > 0$, self-play games are generated:
   i. MCTS samples search probabilities π_t based on the neural network from the previous iteration $f_{\theta_{i-1}}$:

   $$\pi_t = \alpha_{\theta_{i-1}}(s_t)$$

   for each time-step $t = 1, 2, \ldots, T$

   ii. move is sampled from π_t

   iii. data (s_t, π_t, z_t) for each t are stored for later training

   iv. new neural network f_{θ_i} is trained in order to minimize the loss

   $$l = (z - v)^2 - \pi^\top \log \mathbf{p} + c||\theta||^2$$

[Silver et al. 2017b]

AG0: Self-Play Reinforcement Learning – Steps

0. random weights θ_0
1. at each iteration $i > 0$, self-play games are generated:
   i. MCTS samples search probabilities π_t based on the neural network from the previous iteration $f_{\theta_{i-1}}$:

   $$\pi_t = \alpha_{\theta_{i-1}}(s_t)$$

   for each time-step $t = 1, 2, \ldots, T$

   ii. move is sampled from π_t

   iii. data (s_t, π_t, z_t) for each t are stored for later training

   iv. new neural network f_{θ_i} is trained in order to minimize the loss

   $$l = (z - v)^2 - \pi^\top \log \mathbf{p} + c||\theta||^2$$

Loss l makes $(\mathbf{p}, v) = f_\theta(s)$ more closely match the improved search probabilities and self-play winner (π, z).

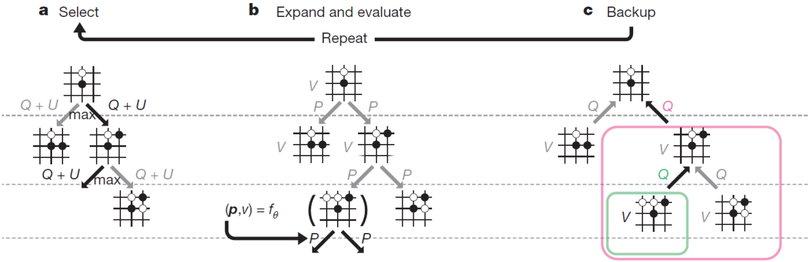
[Silver et al. 2017b]

AG0: Monte Carlo Tree Search (1/2)

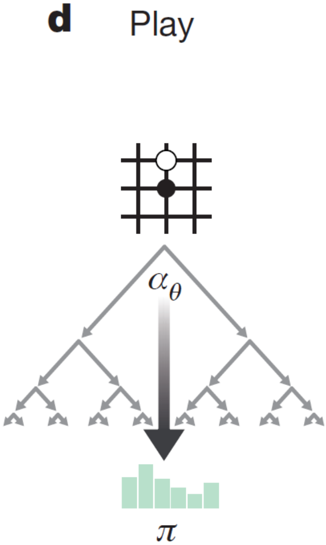
Monte Carlo Tree Search (MCTS) in AG0:

AG0: Monte Carlo Tree Search (1/2)

Monte Carlo Tree Search (MCTS) in AG0:

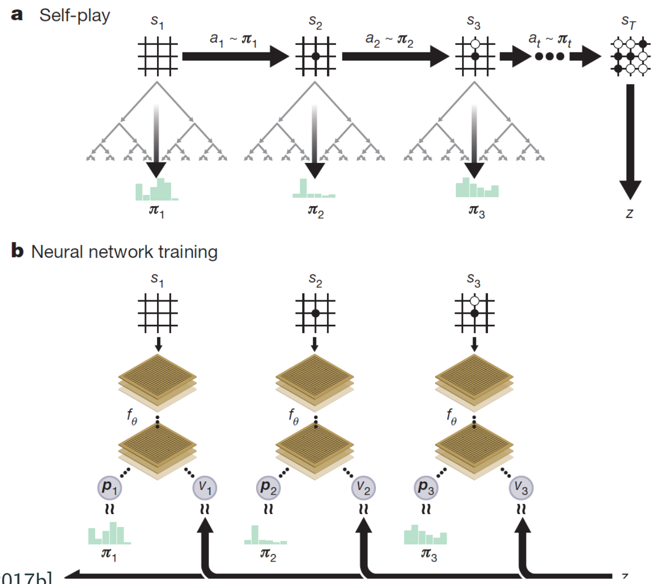


AG0: Monte Carlo Tree Search (2/2)



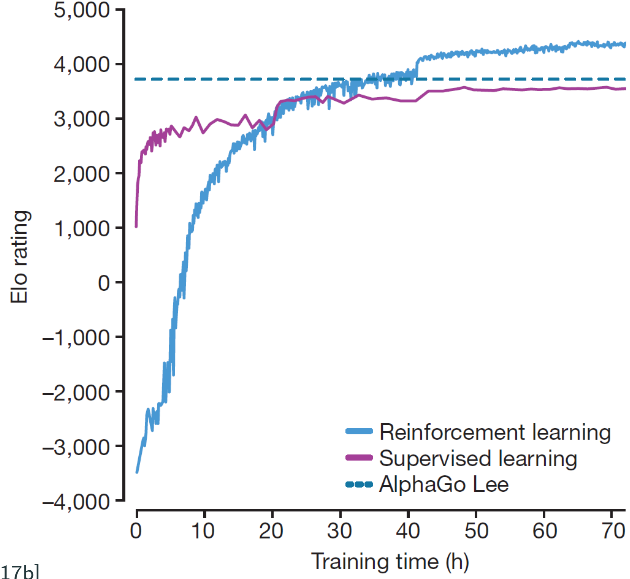
[Silver et al. 2017b]

AG0: Self-Play Reinforcement Learning – Review



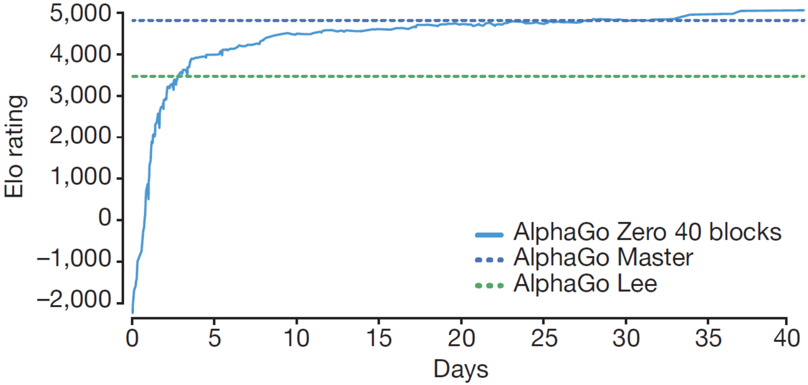
[Silver et al. 2017b]

AG0: Elo Rating over Training Time (RL vs. SL)



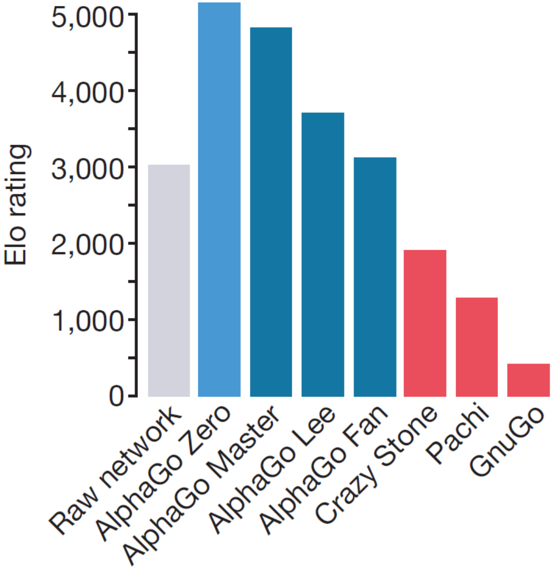
[Silver et al. 2017b]

AG0: Elo Rating over Training Time (AG0 with 40 blocks)



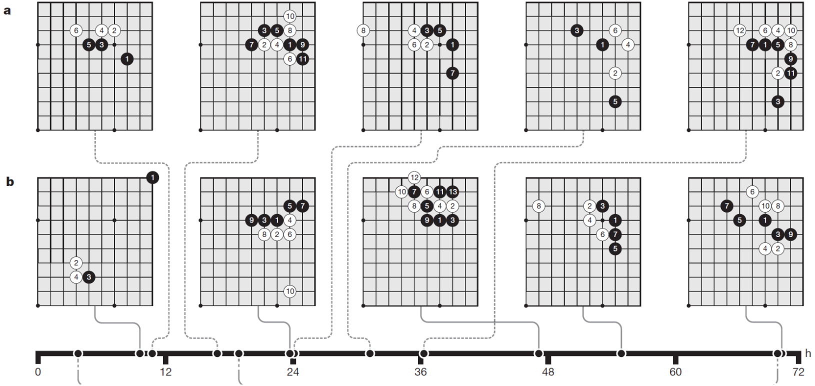
[Silver et al. 2017b]

AG0: Tournament between AI Go Programs



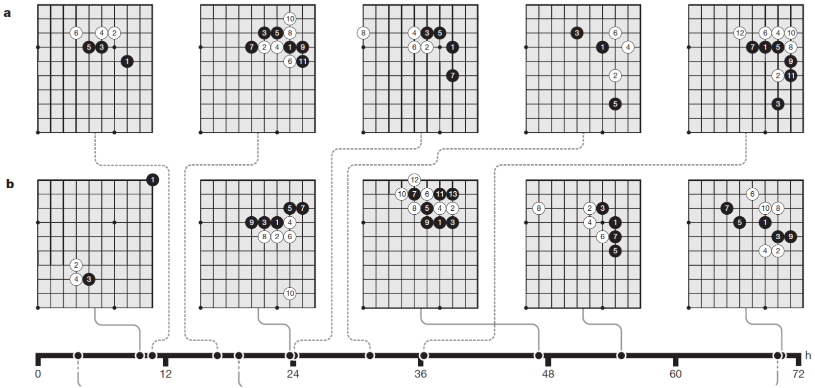
[Silver et al. 2017b]

AG0: Discovered Joseki (Corner Sequences)



[Silver et al. 2017b]

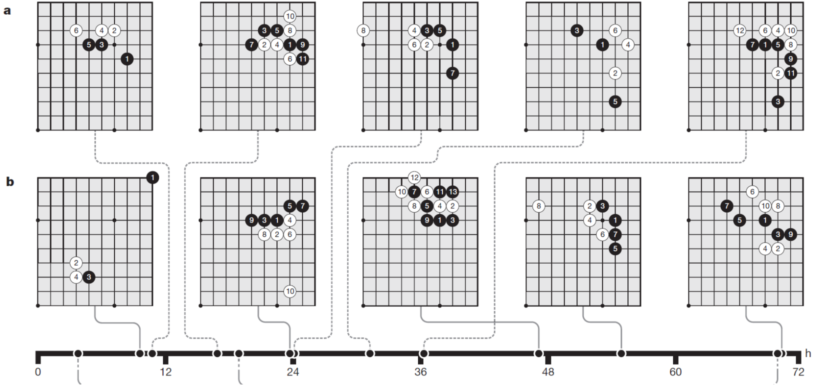
AG0: Discovered Joseki (Corner Sequences)



a five human *joseki*

[Silver et al. 2017b]

AG0: Discovered Joseki (Corner Sequences)

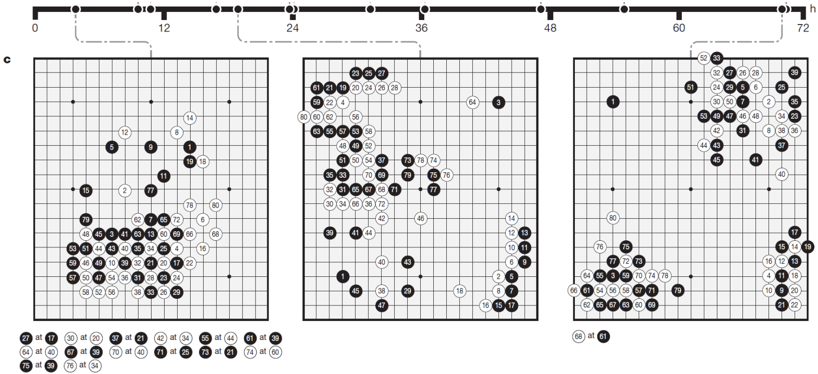


a five human *joseki*

b five novel *joseki* variants eventually preferred by AG0

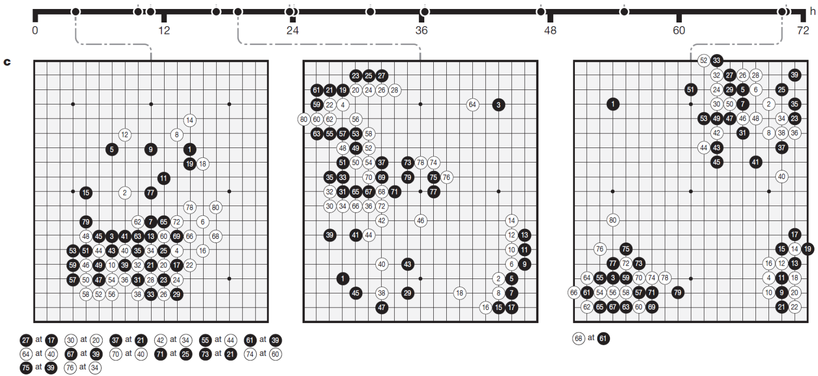
[Silver et al. 2017b]

AG0: Discovered Playing Styles

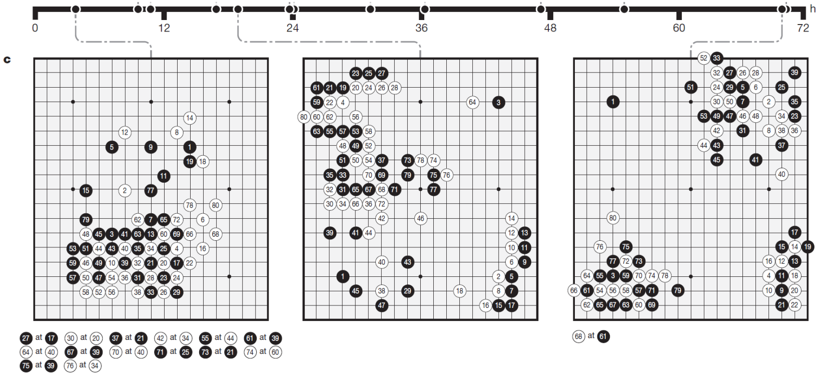


[Silver et al. 2017b]

AG0: Discovered Playing Styles



at 3 h greedy capture of stones

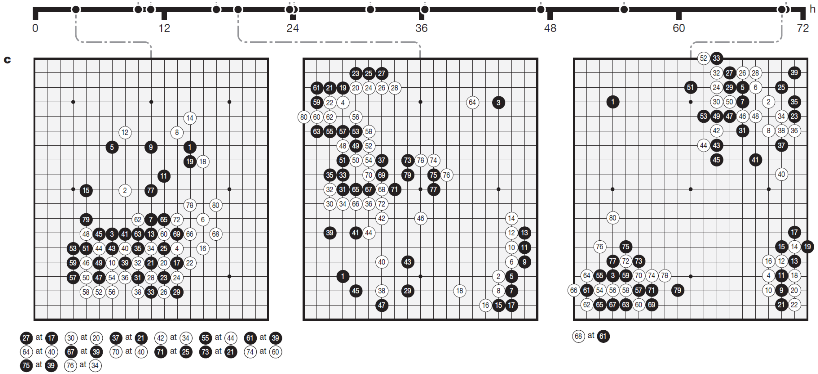


at 3 h greedy capture of stones

at 19 h the fundamentals of Go concepts (life-and-death, influence, territory...)

[Silver et al. 2017b]

AG0: Discovered Playing Styles



at 3 h greedy capture of stones

at 19 h the fundamentals of Go concepts (life-and-death, influence, territory...)

at 70 h remarkably balanced game (multiple battles, complicated *ko* fight, a half-point win for white...)

[Silver et al. 2017b]

AlphaZero

To watch such a strong programme like Stockfish, against whom most top players would be happy to win even one game out of a hundred, being completely taken apart is certainly definitive.

Viswanathan Anand

> To watch such a strong programme like Stockfish, against whom most top players would be happy to win even one game out of a hundred, being completely taken apart is certainly definitive.

Viswanathan Anand

> It's like chess from another dimension.

Demis Hassabis

AlphaZero: Differences Compared to AlphaGo Zero

AlphaGo Zero $\times$ AlphaZero:

AlphaZero: Differences Compared to AlphaGo Zero

AlphaGo Zero $\times$ AlphaZero:

- binary outcome (win / loss) $\times$ expected outcome (including draws or potentially other outcomes)

AlphaZero: Differences Compared to AlphaGo Zero

AlphaGo Zero $\times$ AlphaZero:

- binary outcome (win / loss) $\times$ expected outcome (including draws or potentially other outcomes)
- board positions transformed before passing to neural networks (by randomly selected rotation or reflection) $\times$ no data augmentation

[Silver et al. 2017a]

AlphaZero: Differences Compared to AlphaGo Zero

AlphaGo Zero $\times$ AlphaZero:

- binary outcome (win / loss) $\times$ expected outcome (including draws or potentially other outcomes)
- board positions transformed before passing to neural networks (by randomly selected rotation or reflection) $\times$ no data augmentation
- games generated by the best player from previous iterations (margin of 55 %) $\times$ continual update using the latest parameters (without the evaluation and selection steps)

[Silver et al. 2017a]

AlphaZero: Differences Compared to AlphaGo Zero

AlphaGo Zero $\times$ AlphaZero:

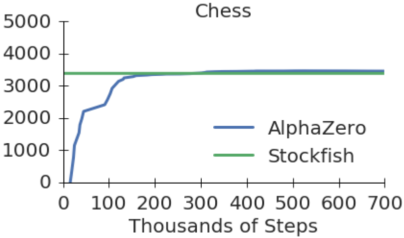
- binary outcome (win / loss) $\times$ expected outcome (including draws or potentially other outcomes)
- board positions transformed before passing to neural networks (by randomly selected rotation or reflection) $\times$ no data augmentation
- games generated by the best player from previous iterations (margin of 55 %) $\times$ continual update using the latest parameters (without the evaluation and selection steps)
- hyper-parameters tuned by Bayesian optimisation $\times$ reused the same hyper-parameters without game-specific tuning

[Silver et al. 2017a]

AlphaZero: Elo Rating over Training Time

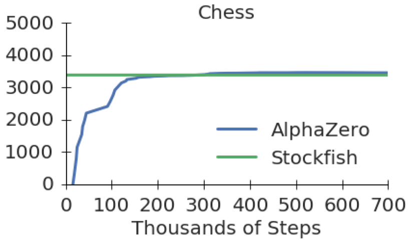
[Silver et al. 2017a]

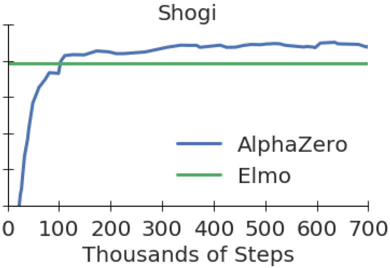
AlphaZero: Elo Rating over Training Time



[Silver et al. 2017a]

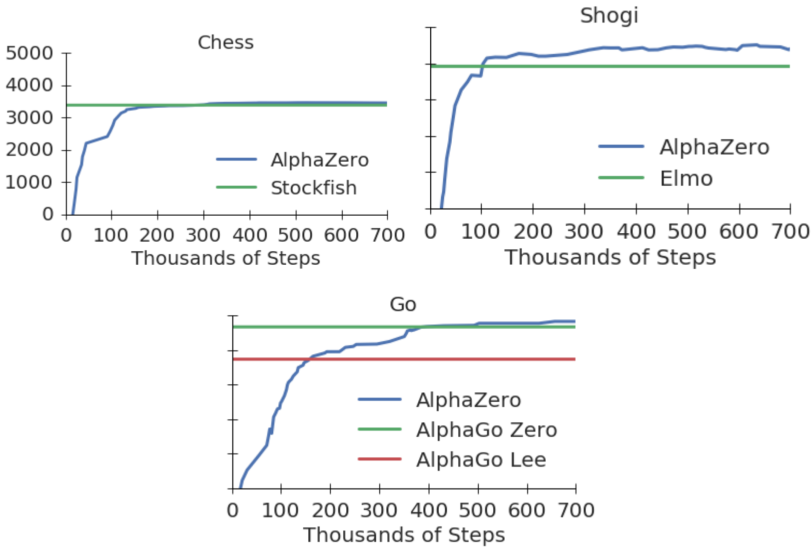
AlphaZero: Elo Rating over Training Time





[Silver et al. 2017a]

AlphaZero: Elo Rating over Training Time



[Silver et al. 2017a]

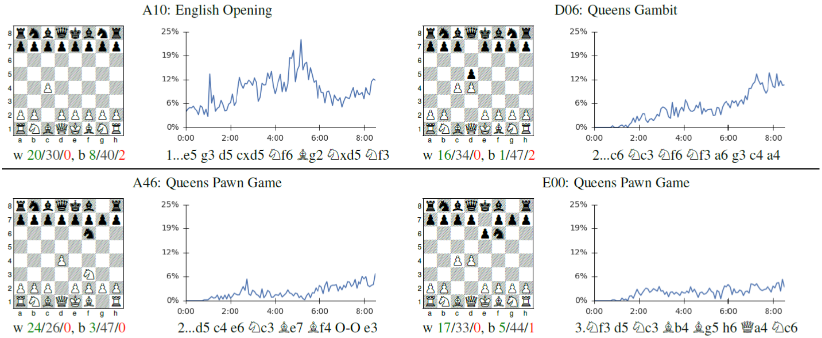
AlphaZero: Tournament between AI Programs

Game	White	Black	Win	Draw	Loss
Chess	*AlphaZero*	*Stockfish*	25	25	0
	Stockfish	*AlphaZero*	3	47	0
Shogi	*AlphaZero*	*Elmo*	43	2	5
	Elmo	*AlphaZero*	47	0	3
Go	*AlphaZero*	*AG0 3-day*	31	–	19
	AG0 3-day	*AlphaZero*	29	–	21

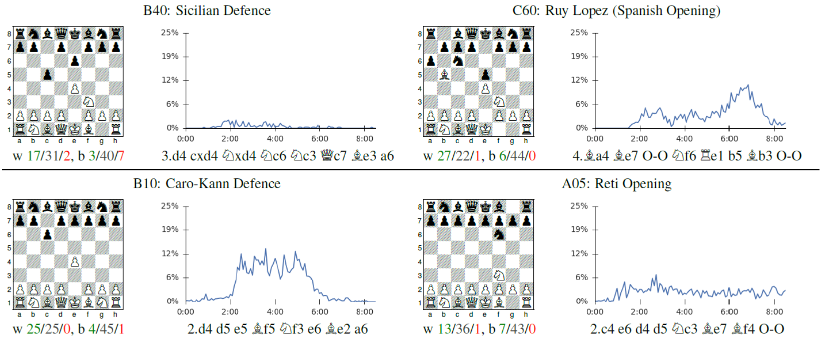
(Values are given from AlphaZero's point of view.)

[Silver et al. 2017a]

AlphaZero: Openings Discovered by the Self-Play (1/2)



AlphaZero: Openings Discovered by the Self-Play (2/2)



Conclusion

Difficulties of Go

- challenging decision-making

Difficulties of Go

- challenging decision-making
- intractable search space

[Silver et al. 2017a]

Difficulties of Go

- challenging decision-making
- intractable search space
- complex optimal solution

  It appears infeasible to directly approximate using a policy or value function!

[Silver et al. 2017a]

AlphaZero: Summary

- Monte Carlo tree search

AlphaZero: Summary

- Monte Carlo tree search
- effective move selection and position evaluation

AlphaZero: Summary

- Monte Carlo tree search
- effective move selection and position evaluation
  - through deep convolutional neural networks

[Silver et al. 2017a]

AlphaZero: Summary

- Monte Carlo tree search
- effective move selection and position evaluation
  - through deep convolutional neural networks
  - trained by new self-play reinforcement learning algorithm

[Silver et al. 2017a]

AlphaZero: Summary

- Monte Carlo tree search
- effective move selection and position evaluation
  - through deep convolutional neural networks
  - trained by new self-play reinforcement learning algorithm
- new search algorithm combining

[Silver et al. 2017a]

AlphaZero: Summary

- Monte Carlo tree search
- effective move selection and position evaluation
  - through deep convolutional neural networks
  - trained by new self-play reinforcement learning algorithm
- new search algorithm combining
  - evaluation by a single neural network

[Silver et al. 2017a]

AlphaZero: Summary

- Monte Carlo tree search
- effective move selection and position evaluation
  - through deep convolutional neural networks
  - trained by new self-play reinforcement learning algorithm
- new search algorithm combining
  - evaluation by a single neural network
  - Monte Carlo tree search

[Silver et al. 2017a]

AlphaZero: Summary

- Monte Carlo tree search
- effective move selection and position evaluation
  - through deep convolutional neural networks
  - trained by new self-play reinforcement learning algorithm
- new search algorithm combining
  - evaluation by a single neural network
  - Monte Carlo tree search
- more efficient when compared to previous AlphaGo versions

[Silver et al. 2017a]

AlphaZero: Summary

- Monte Carlo tree search
- effective move selection and position evaluation
  - through deep convolutional neural networks
  - trained by new self-play reinforcement learning algorithm
- new search algorithm combining
  - evaluation by a single neural network
  - Monte Carlo tree search
- more efficient when compared to previous AlphaGo versions
  - single machine

[Silver et al. 2017a]

AlphaZero: Summary

- Monte Carlo tree search
- effective move selection and position evaluation
  - through deep convolutional neural networks
  - trained by new self-play reinforcement learning algorithm
- new search algorithm combining
  - evaluation by a single neural network
  - Monte Carlo tree search
- more efficient when compared to previous AlphaGo versions
  - single machine
  - 4 TPUs

[Silver et al. 2017a]

AlphaZero: Summary

- Monte Carlo tree search
- effective move selection and position evaluation
  - through deep convolutional neural networks
  - trained by new self-play reinforcement learning algorithm
- new search algorithm combining
  - evaluation by a single neural network
  - Monte Carlo tree search
- more efficient when compared to previous AlphaGo versions
  - single machine
  - 4 TPUs
  - hours rather than months of training time

[Silver et al. 2017a]

Novel approach

[Silver et al. 2017a]

Novel approach

During the matches (against Stockfish and Elmo), AlphaZero evaluated thousands of times fewer positions than Deep Blue against Kasparov.

[Silver et al. 2017a]

Novel approach

During the matches (against Stockfish and Elmo), AlphaZero evaluated thousands of times fewer positions than Deep Blue against Kasparov.

It compensated this by:

- selecting those positions more intelligently (the neural network)

[Silver et al. 2017a]

Novel approach

During the matches (against Stockfish and Elmo), AlphaZero evaluated thousands of times fewer positions than Deep Blue against Kasparov.

It compensated this by:

- selecting those positions more intelligently (the neural network)
- evaluating them more precisely (the same neural network)

[Silver et al. 2017a]

Novel approach

During the matches (against Stockfish and Elmo), AlphaZero evaluated thousands of times fewer positions than Deep Blue against Kasparov.

It compensated this by:

- selecting those positions more intelligently (the neural network)
- evaluating them more precisely (the same neural network)

[Silver et al. 2017a]

Novel approach

During the matches (against Stockfish and Elmo), AlphaZero evaluated thousands of times fewer positions than Deep Blue against Kasparov.

It compensated this by:

- selecting those positions more intelligently (the neural network)
- evaluating them more precisely (the same neural network)

Deep Blue relied on a handcrafted evaluation function.

[Silver et al. 2017a]

Novel approach

During the matches (against Stockfish and Elmo), AlphaZero evaluated thousands of times fewer positions than Deep Blue against Kasparov.

It compensated this by:

- selecting those positions more intelligently (the neural network)
- evaluating them more precisely (the same neural network)

Deep Blue relied on a handcrafted evaluation function.

AlphaZero was trained tabula rasa from self-play. It used general-purpose learning.

[Silver et al. 2017a]

Novel approach

During the matches (against Stockfish and Elmo), AlphaZero evaluated thousands of times fewer positions than Deep Blue against Kasparov.

It compensated this by:

- selecting those positions more intelligently (the neural network)
- evaluating them more precisely (the same neural network)

Deep Blue relied on a handcrafted evaluation function.

AlphaZero was trained tabula rasa from self-play. It used general-purpose learning.

This approach is not specific to the game of Go. The algorithm can be used for much wider class of AI problems!

[Silver et al. 2017a]

Thank you!

Questions?

Backup Slides

Input Features of AlphaZero's Neural Networks

Go		Chess		Shogi	
Feature	Planes	Feature	Planes	Feature	Planes
P1 stone	1	P1 piece	6	P1 piece	14
P2 stone	1	P2 piece	6	P2 piece	14
		Repetitions	2	Repetitions	3
				P1 prisoner count	7
				P2 prisoner count	7
Colour	1	Colour	1	Colour	1
		Total move count	1	Total move count	1
		P1 castling	2		
		P2 castling	2		
		No-progress count	1		
Total	17	Total	119	Total	362

[Silver et al. 2017a]

AlphaZero: Statistics of Training

	Chess	Shogi	Go
Mini-batches	700k	700k	700k
Training Time	9h	12h	34h
Training Games	44 million	24 million	21 million
Thinking Time	800 sims	800 sims	800 sims
	40 ms	80 ms	200 ms

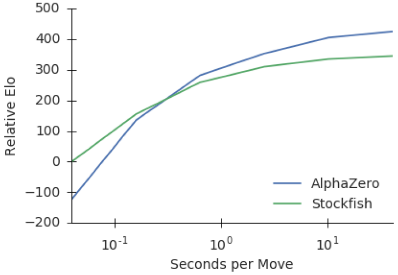
[Silver et al. 2017a]

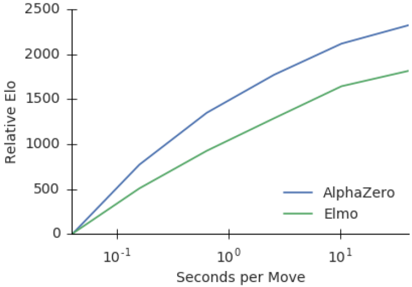
AlphaZero: Evaluation Speeds

Program	Chess	Shogi	Go
AlphaZero	80k	40k	16k
Stockfish	70,000k		
Elmo		35,000k	

[Silver et al. 2017a]

Scalability When Compared to Other Programs





[Silver et al. 2017a]

Further Reading I

AlphaGo:

- **Google Research Blog**
  http://googleresearch.blogspot.cz/2016/01/alphago-mastering-ancient-game-of-go.html
- an article in **Nature**
  http://www.nature.com/news/google-ai-algorithm-masters-ancient-game-of-go-1.19234
- a **reddit** article claiming that AlphaGo is even stronger than it appears to be:
  "AlphaGo would rather win by less points, but with higher probability."
  https://www.reddit.com/r/baduk/comments/49y17z/the_true_strength_of_alphago/
- a video of how AlphaGo works (put in layman's terms) https://youtu.be/qWcfiPi9gUU

Articles by Google DeepMind:

- **Atari player**: a DeepRL system which combines Deep Neural Networks with Reinforcement Learning (Mnih et al. 2015)
- **Neural Turing Machines** (Graves, Wayne, and Danihelka 2014)

Artificial Intelligence:

- **Artificial Intelligence course at MIT**
  http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-034-artificial-intelligence-fall-2010/index.htm

Further Reading II

- **Introduction to Artificial Intelligence at Udacity** `https://www.udacity.com/course/intro-to-artificial-intelligence--cs271`
- **General Game Playing course** `https://www.coursera.org/course/ggp`
- **Singularity** `http://waitbutwhy.com/2015/01/artificial-intelligence-revolution-1.html` + Part 2
- **The Singularity Is Near** (Kurzweil 2005)

Combinatorial Game Theory (founded by John H. Conway to study endgames in Go):

- **Combinatorial Game Theory course** `https://www.coursera.org/learn/combinatorial-game-theory`
- On Numbers and Games (Conway 1976)
- Computer Go as a sum of local games: an application of combinatorial game theory (Müller 1995)

Chess:

- **Deep Blue beats G. Kasparov in 1997** `https://youtu.be/NJarxpYyoFI`

Machine Learning:

- **Machine Learning course** `https://youtu.be/hPKJBXkyTK://www.coursera.org/learn/machine-learning/`
- **Reinforcement Learning** `http://reinforcementlearning.ai-depot.com/`
- **Deep Learning** (LeCun, Bengio, and Hinton 2015)

Further Reading III

- **Deep Learning course** https://www.udacity.com/course/deep-learning--ud730
- **Two Minute Papers** https://www.youtube.com/user/keeroyz
- **Applications of Deep Learning** https://youtu.be/hPKJBXkyTKM

Neuroscience:

- http://www.brainfacts.org/

References I

Conway, John Horton (1976). "On Numbers and Games". In: *London Mathematical Society Monographs* 6.

Graves, Alex, Greg Wayne, and Ivo Danihelka (2014). "Neural Turing Machines". In: *arXiv preprint arXiv:1410.5401*.

Kurzweil, Ray (2005). *The Singularity is Near: When Humans Transcend Biology*. Penguin.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep Learning". In: *Nature* 521.7553, pp. 436–444.

Mnih, Volodymyr et al. (2015). "Human-Level Control through Deep Reinforcement Learning". In: *Nature* 518.7540, pp. 529–533. URL: `https://storage.googleapis.com/deepmind-data/assets/papers/DeepMindNature14236Paper.pdf`.

Müller, Martin (1995). "Computer Go as a Sum of Local Games: an Application of Combinatorial Game Theory". PhD thesis. TU Graz.

Silver, David et al. (2016). "Mastering the Game of Go with Deep Neural Networks and Tree Search". In: *Nature* 529.7587, pp. 484–489.

Silver, David et al. (2017a). "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm". In: *arXiv preprint arXiv:1712.01815*.

Silver, David et al. (2017b). "Mastering the Game of Go without Human Knowledge". In: *Nature* 550.7676, pp. 354–359.