style transfer

CSCI 381

chess without search

Alex K.

Final Project

CSCI 381

28 years

~~not too~~ long ago


the rematch

nowadays...

WHO WOULD WIN?

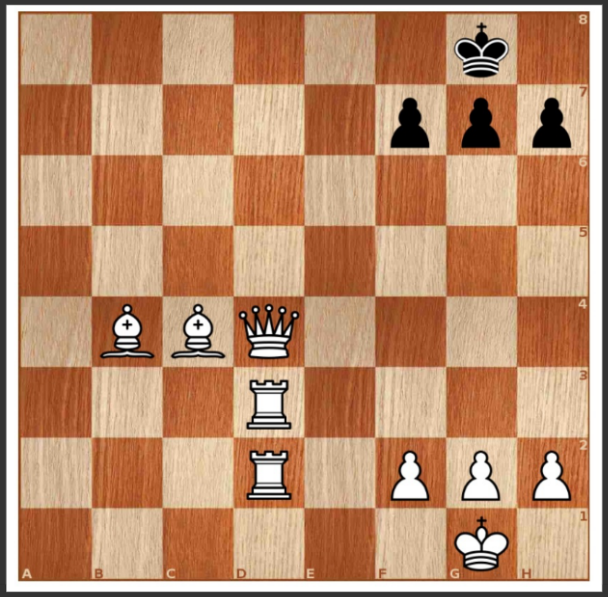
The world's strongest chess players

One fishy boi (he does minimax search really fast)

Let's make a computer play chess

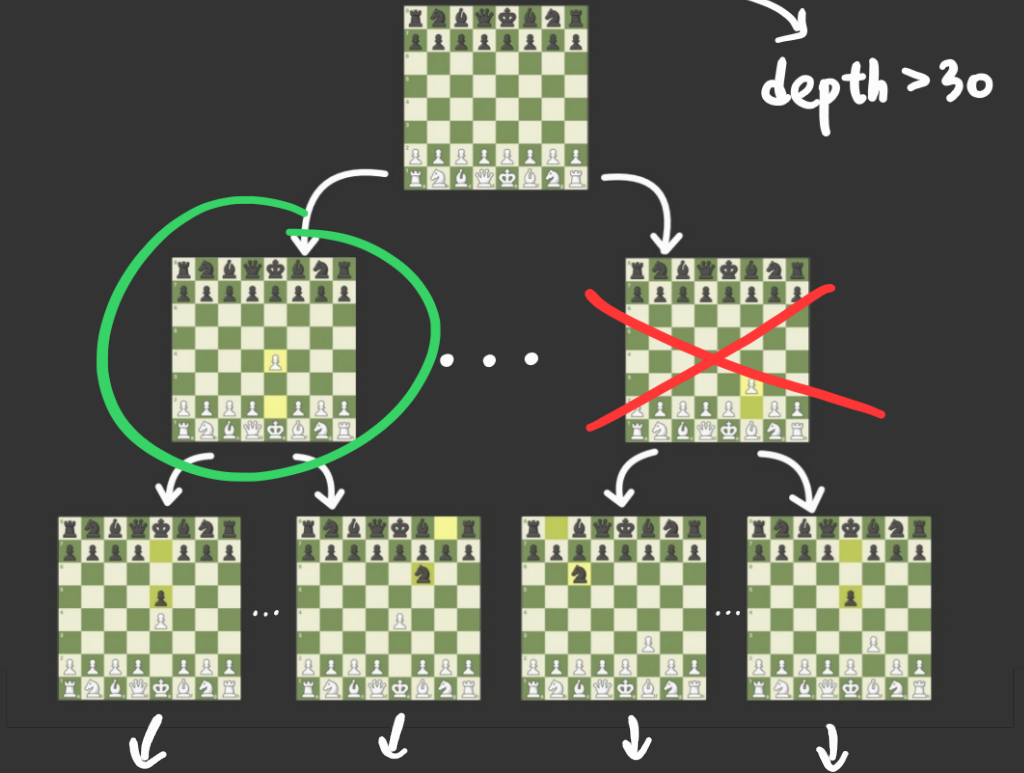
idea:

evaluate board positions



Searching deep is key

depth > 30



unless you don't?

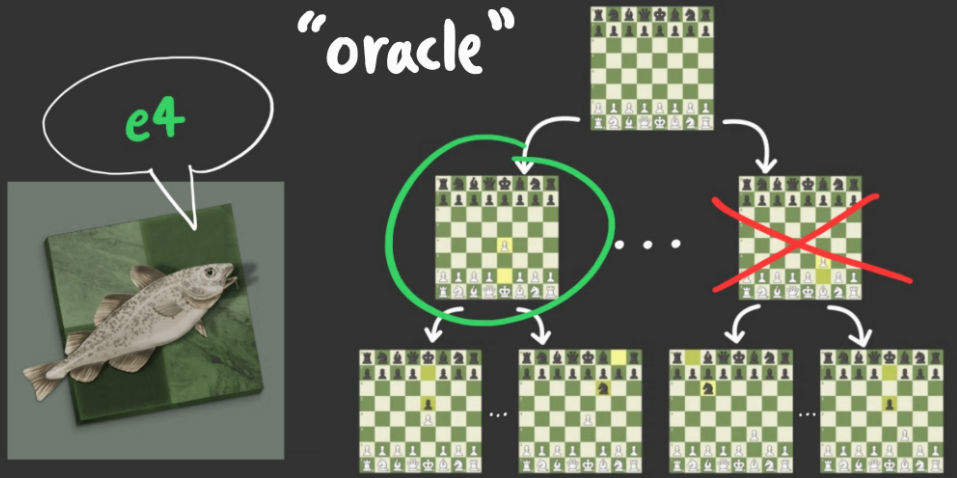
Google DeepMind

Grandmaster-Level Chess Without Search

Anian Ruoss[*,1], Grégoire Delétang[*,1], Sourabh Medapati[1], Jordi Grau-Moya[1], Li Kevin Wenliang[1], Elliot Catt[1], John Reid[1] and Tim Genewein[1]
[*]Equal contributions, [1]Google DeepMind

The recent breakthrough successes in machine learning are mainly attributed to scale: namely large-scale attention-based architectures and datasets of unprecedented scale. This paper investigates the impact of training at scale for chess. Unlike traditional chess engines that rely on complex heuristics, explicit search, or a combination of both, we train a 270M parameter transformer model with supervised learning on a dataset of 10 million chess games. We annotate each board in the dataset with action-values provided by the powerful Stockfish 16 engine, leading to roughly 15 billion data points. Our largest model reaches a Lichess blitz Elo of 2895 against humans, and successfully solves a series of challenging chess puzzles, without any domain-specific tweaks or explicit search algorithms. We also show that our model outperforms AlphaZero's policy and value networks (without MCTS) and GPT-3.5-turbo-instruct. A systematic investigation of model and dataset size shows that strong chess performance only arises at sufficient scale. To validate our results, we perform an extensive series of ablations of design choices and hyperparameters.

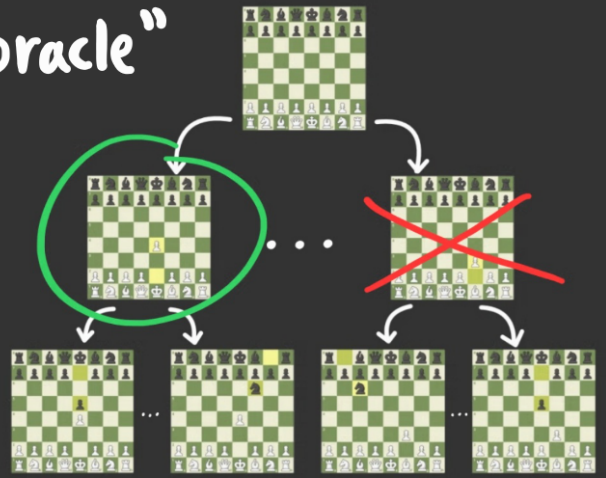
learn to copy Stockfish to



learn to copy Stockfish to

evaluate **moves** not boards

"oracle"

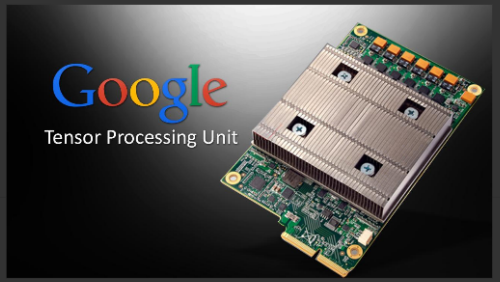


e4 is good given

tokenize 

explicit search, or a combination of both, we train a 270M parameter transformer model with supervised learning on a dataset of 10 million chess games. We annotate each board in the dataset with action-values provided by the powerful Stockfish 16 engine, leading to roughly 15 billion data points. Our largest

Training protocol Predictors are trained by minimizing cross-entropy loss (i.e., log-loss) via mini-batch based stochastic gradient descent using Adam (Kingma and Ba, 2015). We train for 10 million steps, which corresponds to 2.67 epochs for a batch



Can you do that for ~15B more moves

While Adam does its thing

Thx

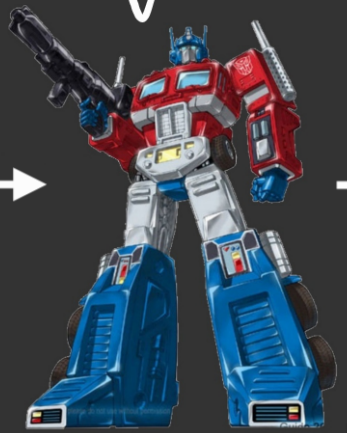
generalize to unseen positions at test time

posterior probability distribution

Input

what would  have done here...

tokenize 



Output

Move	P(win)
f3+	Bin: >99%
Rxe1	Bin: 61~62%
⋮	⋮
Qxg4+	Bin: <1%

it plays well

Feb 2024

model reaches a Lichess blitz Elo of 2895 against humans,

← beats 99.99% of human players

← beats every human easily

Mar 2024

it distills an "approximation" of its oracle

We distill an approximation of Stockfish 16 into a neural predictor that generalizes well to novel board states.

=

low res (

)

but what does that mean

6. Conclusion

Our paper shows that is is possible to distill an approximation of Stockfish 16 into a feed-forward transformer via standard supervised training. The resulting predictor generalizes well to unseen board states, and, when used in a policy, leads to strong chess play (Lichess Elo of 2895 against humans). We demonstrate that strong chess capabilities from supervised learning only emerge at sufficient dataset and model scale. Our work thus adds to a rapidly growing body of literature showing that complex and sophisticated algorithms can be distilled into feed-forward transformers, implying a paradigm-shift away from viewing large transformers as "mere" statistical pattern recognizers to viewing them as a powerful technique for general algorithm approximation.

we train a 270M parameter transformer model

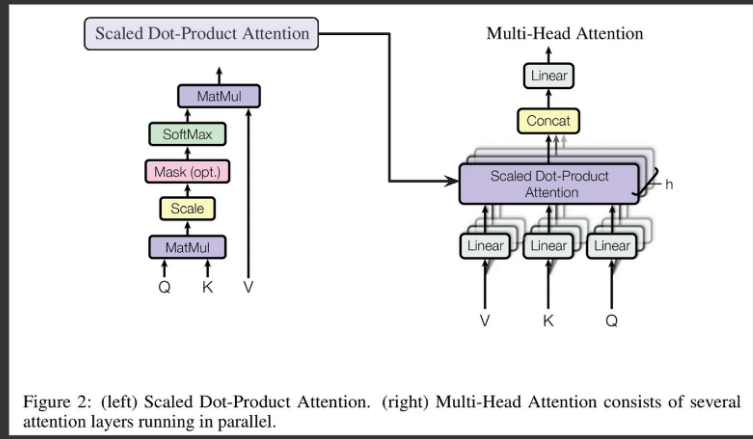


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

← from "Attention is all you need"

Somewhere in there is an approximated search with domain knowledge

Thanks