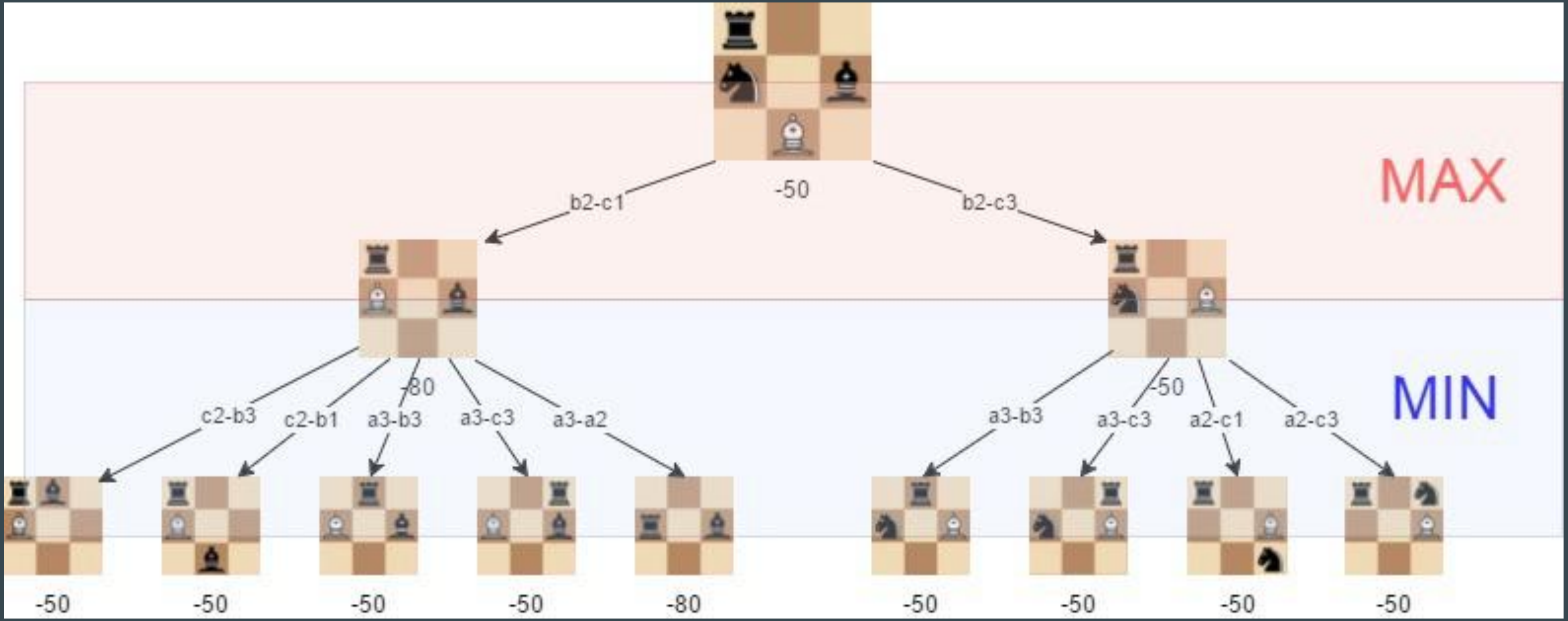
DeepLew: My Chess Engine

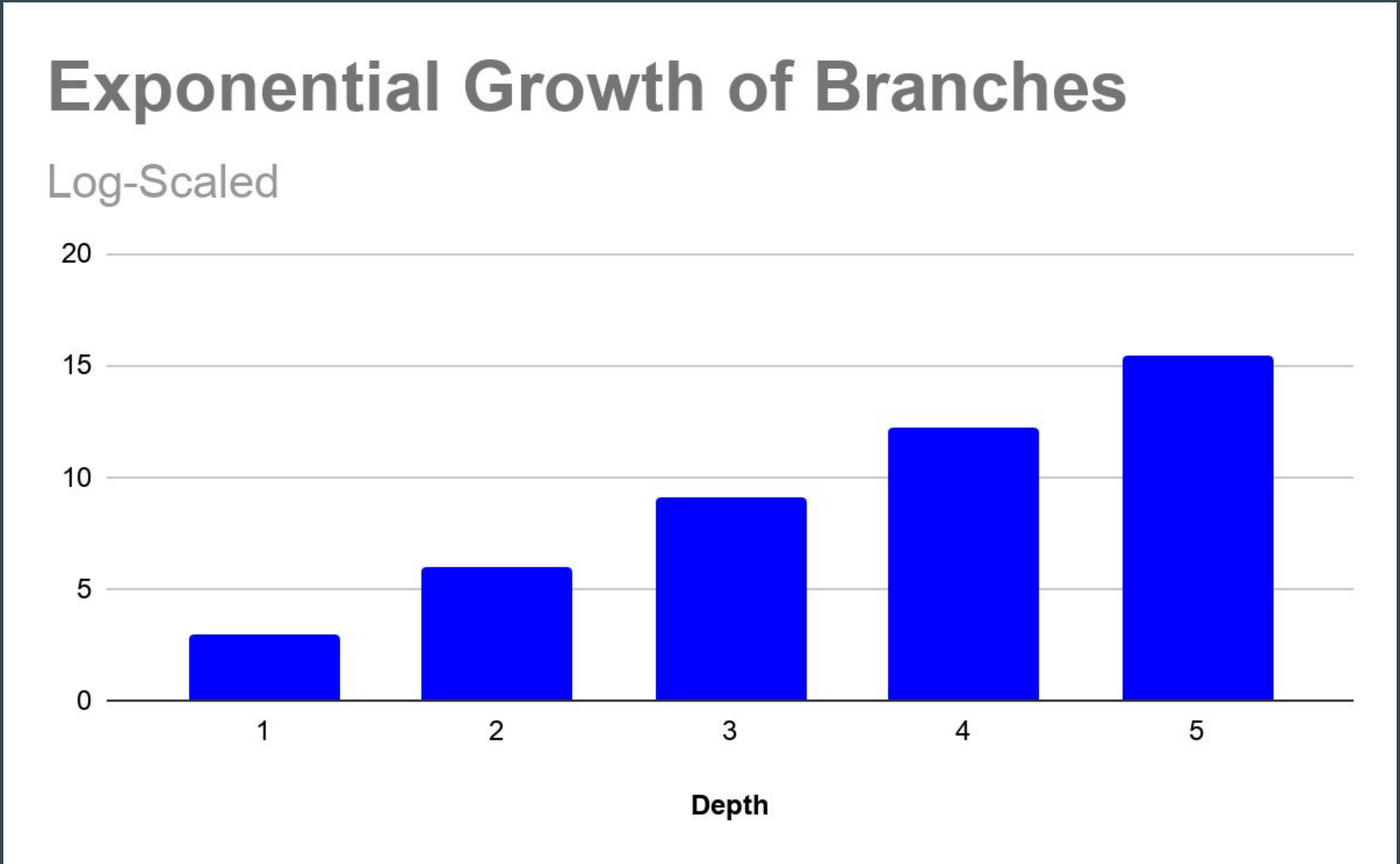
•••

Lew Sears IV
Metis 2020

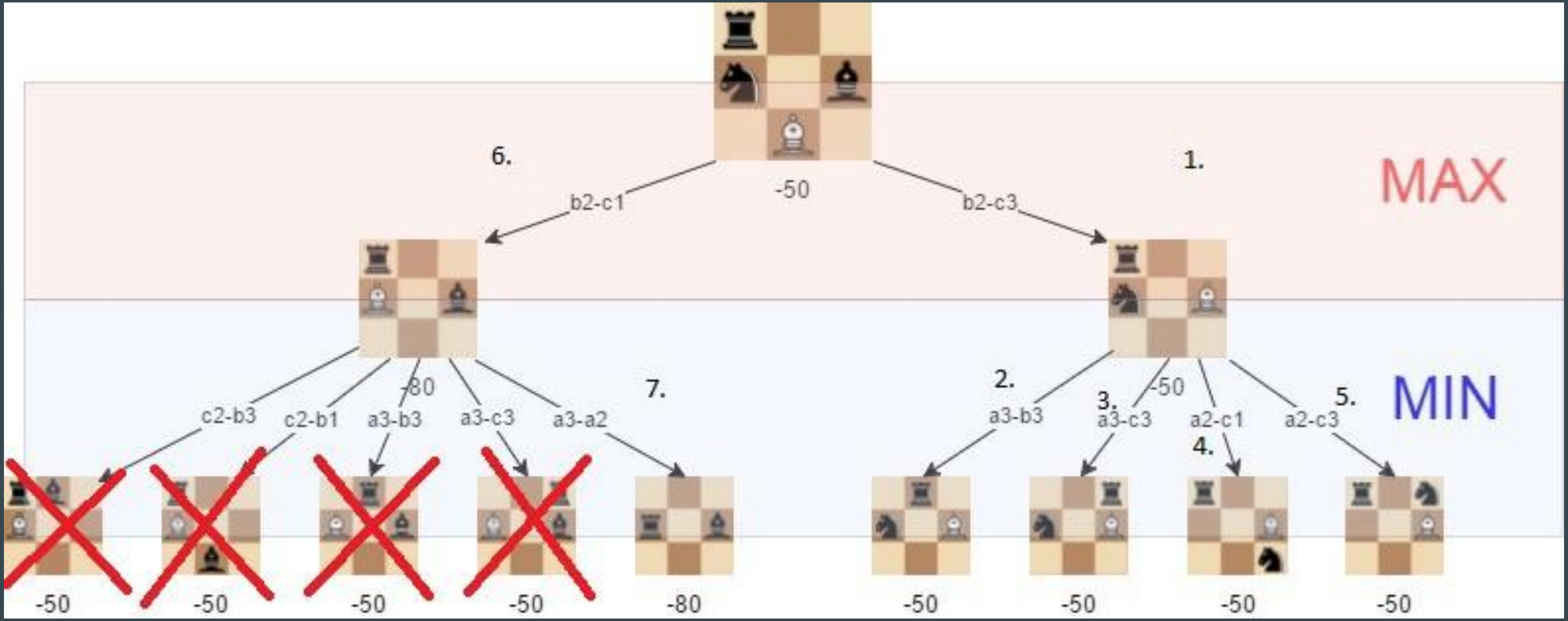
IBM’s Deep Blue: 1996

Initial Algorithm: MiniMax Algorithm

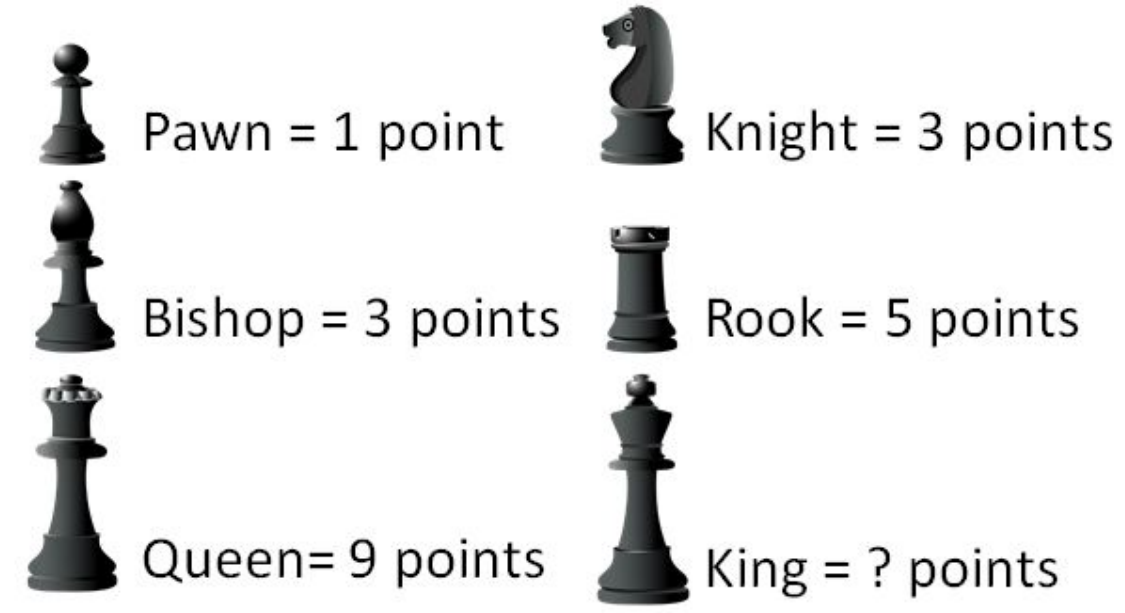



Exponential Growth of Branches
Log-Scaled
20
15
10
5
0
1
2
3
4
5
Depth

Alpha-Beta Pruning to Increase Speed



DeepLew: First Iteration



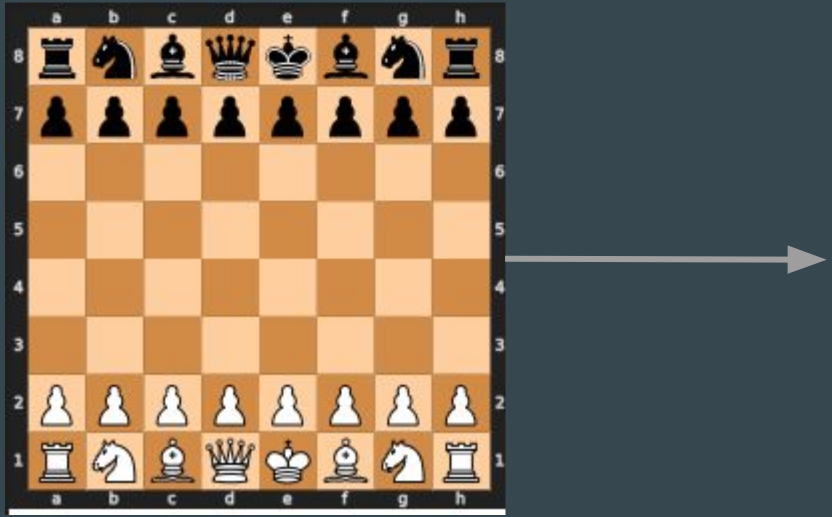
- Recursive MiniMax Algorithm with Alpha Beta Pruning
- Calculate net points by capture/loss of pieces
- Incentivize moving pieces to central squares
- Incentivize Castling

Incorporating Neural Networks

DeepMind’s AlphaZero: 2017

DeepLew: Second Iteration

- Train a Convolutional Neural Network on Professional Games
- 20,000 Games
  - 1.2 Million Board States
- Input board state
- Output Move



“E4”

The Design

a
b
c
d
e
f
g
h
8
7
6
5
4
3
2
1

- Create a one-hot encoded board
  - 9 x 8 x 12 Tensor
  - 8 x 8 board and 12 pieces
  - Extra dimension for white/black
- Train a 2D CNN with 4 layers
- Outputs 27-dimensional move vector
  - Encoded piece type and coordinate

"E4"

DeepLew: Second Iteration

Successes:

- Makes theoretically sound moves Minimax lacks the depth to make

Complications:

- Occasionally makes impossible moves
- Unique boards create problems

DeepLew: Third Iteration

Hybrid Model

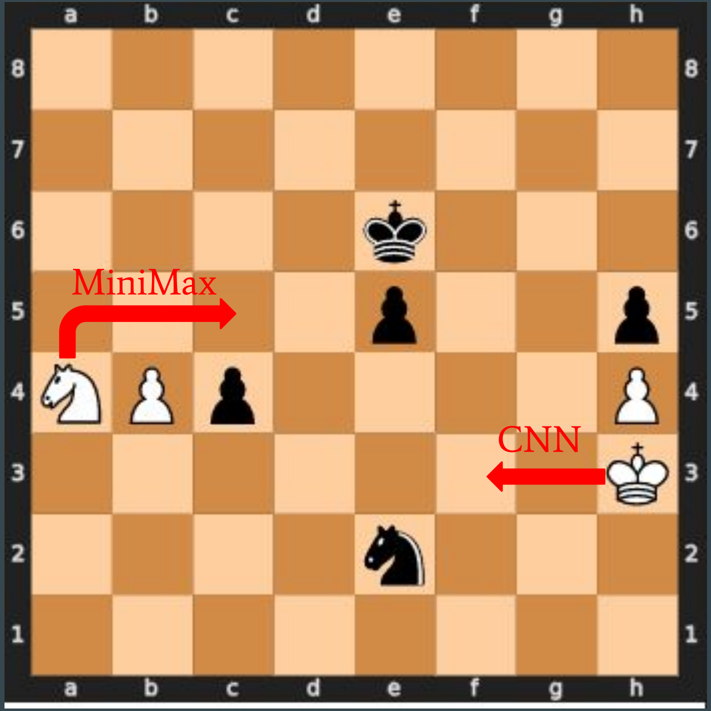
- Make a move based on CNN prediction
- If the move is impossible, revert to minimax

Working Idea:

- Set max moves for CNN

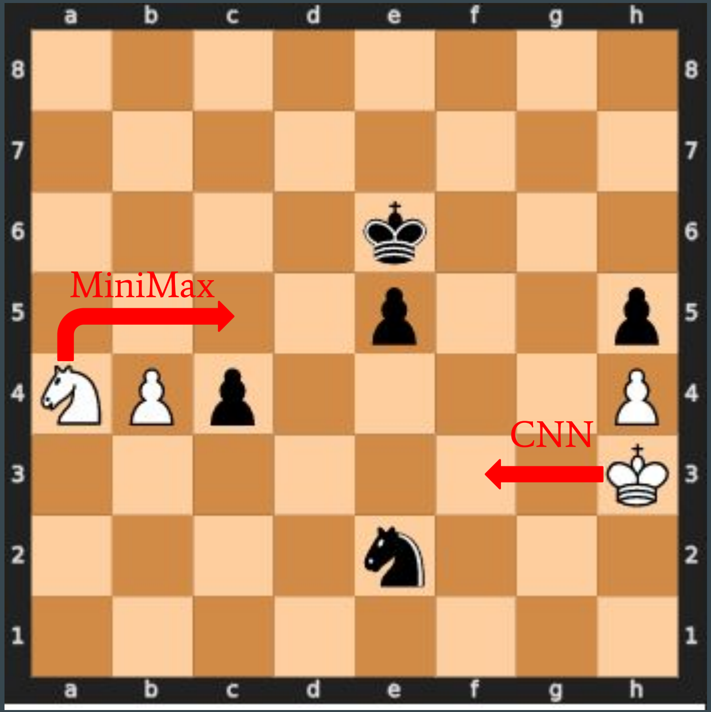
DeepLew Plays Lew: The Sicilian

Unique Board Error: Fischer vs. Larson



Unique Board Error: Fischer vs. Larson

- Famous game from 1967
- Currently on move 56
- CNN predicts ‘Kf3’
  - Not a possible move
- Revert to Minimax
  - At depth 6, plays ‘Nc5+’
  - Makes the same move as Danish grandmaster Bent Larson



Future DeepLew: Reinforced Learning

Thank you!

Please feel free to reach out with any questions!

- LinkedIn
- GitHub
- Email: lewissearsiv@gmail.com