Dynamic Chess Valuation with Deep Neural Networks



Travis J. Burrows, Gabriel Nakajima An, Alen Polakof, Jeyanth John Britto, Mihir Tulpule, & Pranshav Thakkar



Motivation

- » Chess engines solely provide the optimal move, which is not helpful for players trying to analyse and improve upon their game
- »Our tool provides chess players with a user-friendly platform and an easily interpretable metric: piece value

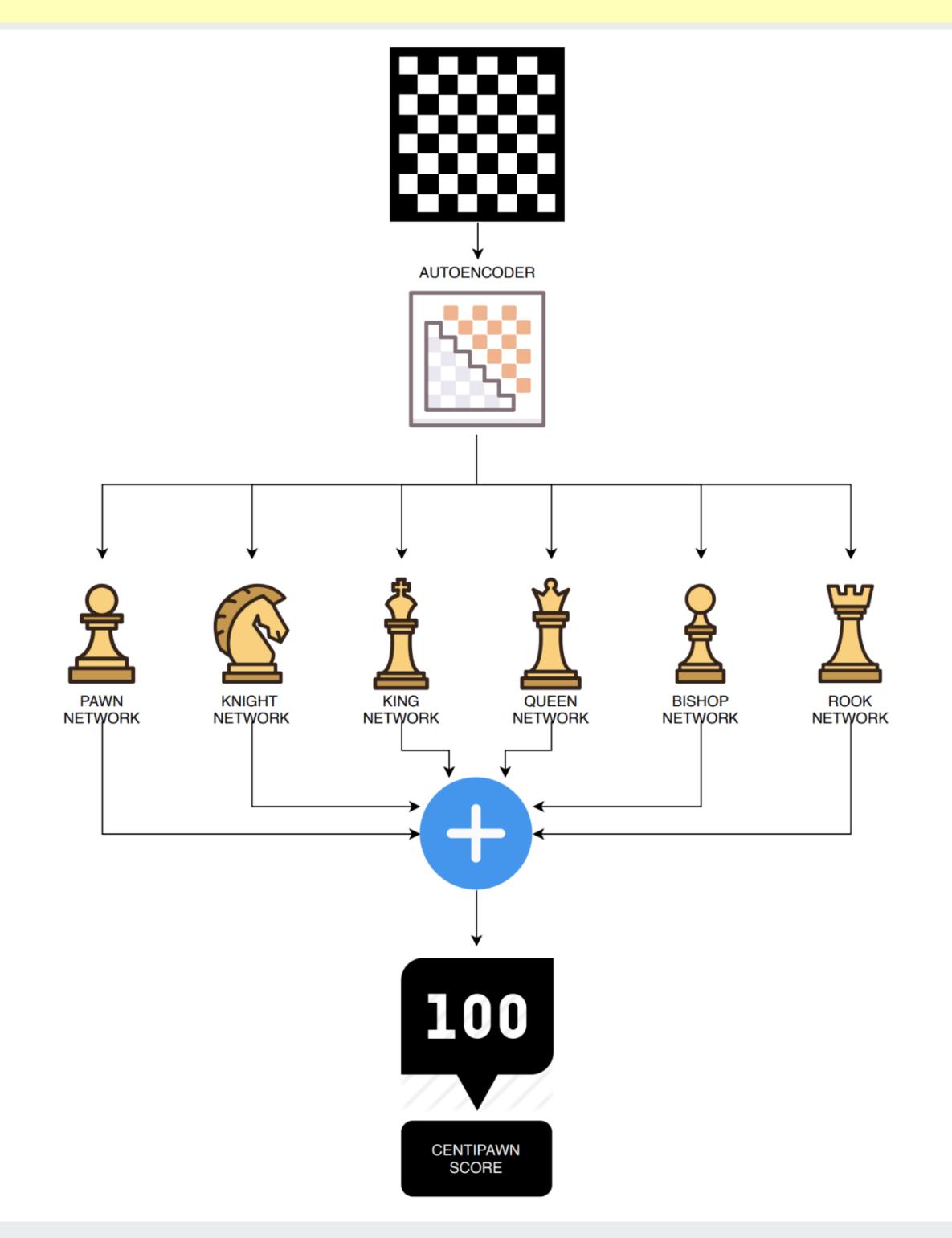
Approach

- » Machine learning model: use autoencoder and multiple neural networks to predict chess piece values based on board configuration
 - PyTorch, Stockfish
- »User interface: Interactive chess board to indicate changing piece values based on a given board configuration
 - Chess.js, Chessboard.js, Chart.js, D3, Flask

Data

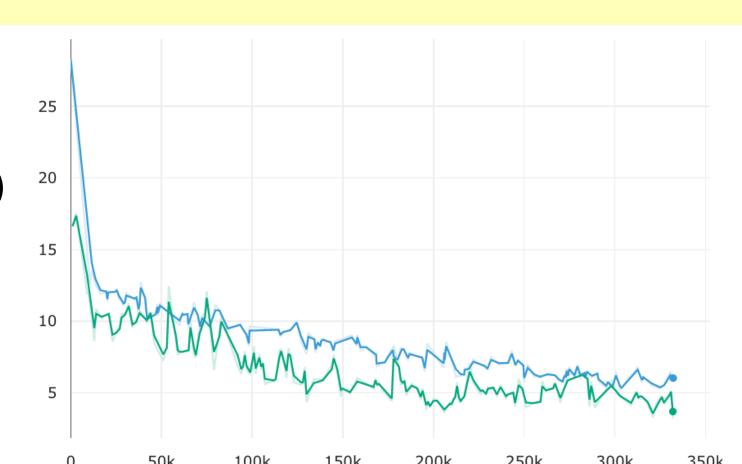
- » Chess game records were downloaded online and for each one, metadata was generated by utilizing Stockfish chess engine
- »1.1 million boards from 6 million total chess game configurations
 - Includes board configuration, centipawn score, metadata
- »Total dataset is 1.2 GB in size

Model



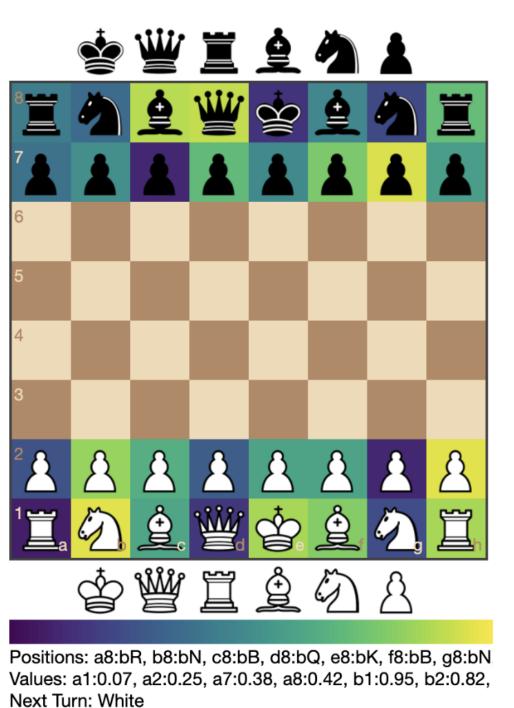
Experiments

- » Centipawn (1/100 of a pawn) score from Stockfish chess engine was used as the response for each configuration to measure player advantage (rather than just a binary winning/losing metric)
- » Centipawn score is an expectation of the margin by which a player is winning, according to the positioning of the pieces
- » Different-size datasets and training times were tested for higher accuracy in the model



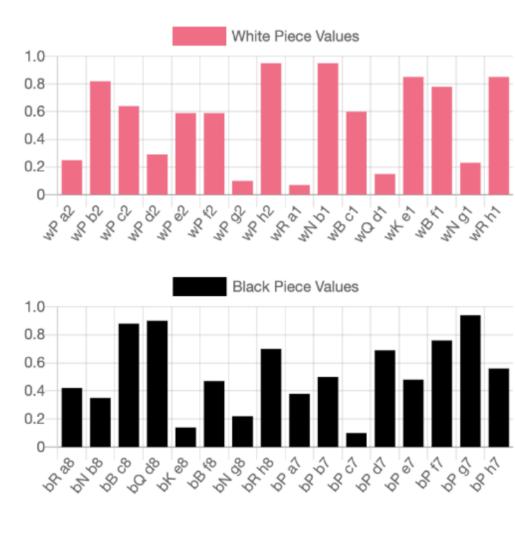
Piece	Traditional	Our Model
Pawn	1.00	1.00
Knight	3.00	2.94
Bishop	3.00	3.13
Rook	5.00	4.96
Queen	9.00	9.87

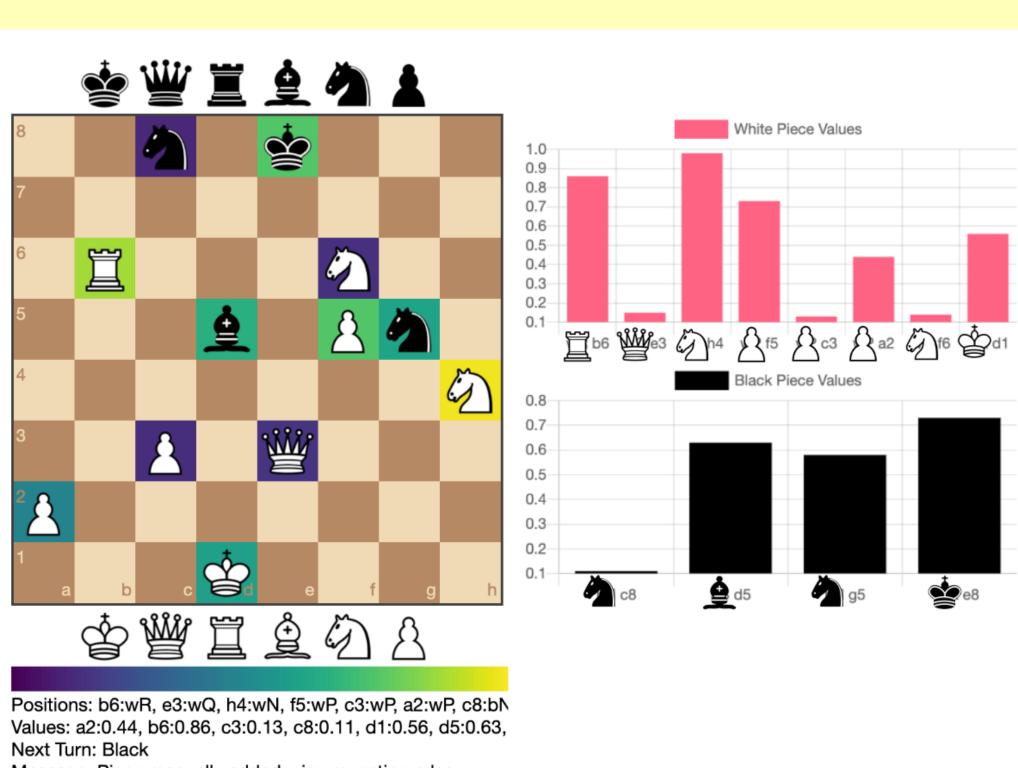
Result: Informative Tool with Interactive Visualization



Message: Board and game reset to start

Clear Board Start Position





Message: Piece manually added, circumventing rules

Clear Board Start Position