Predicting Chess Game Winner using Neural Networks

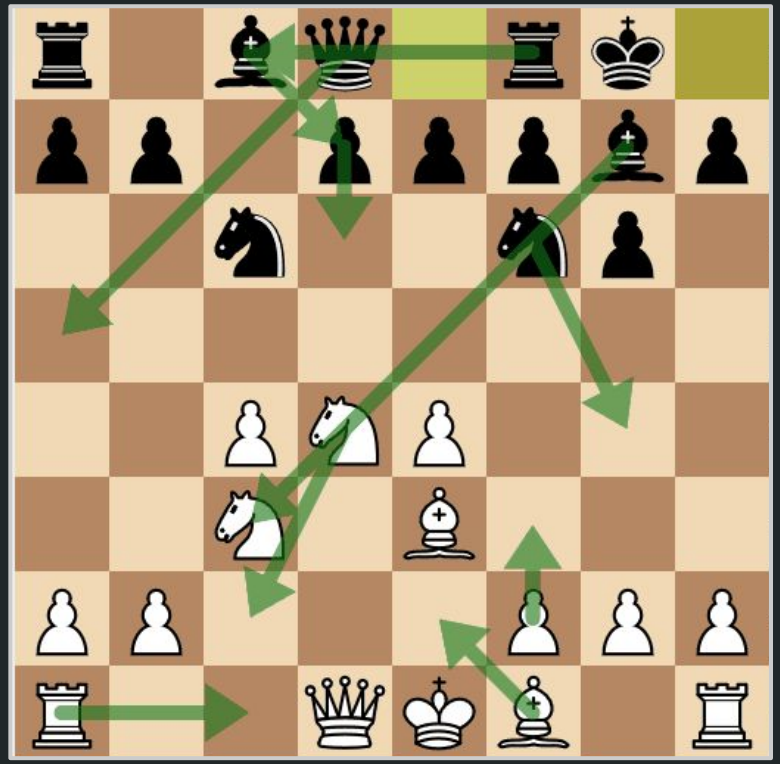
Clement Gendler

How accurately can the outcome of a chess game (winner or loser) be predicted by a neural network using ***only*** the moves played?

Why only use the moves played?

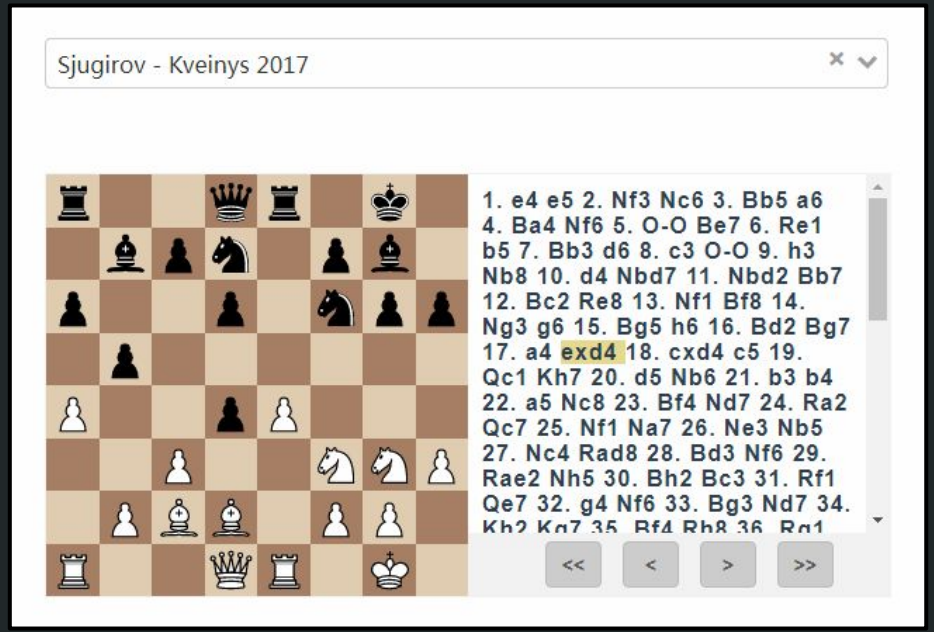
Significance of this Topic

- Creating a nuanced and unique neural network
- Identify critical positions and mistakes ingame
- Provides insight into moves and sequences that are effective



Data

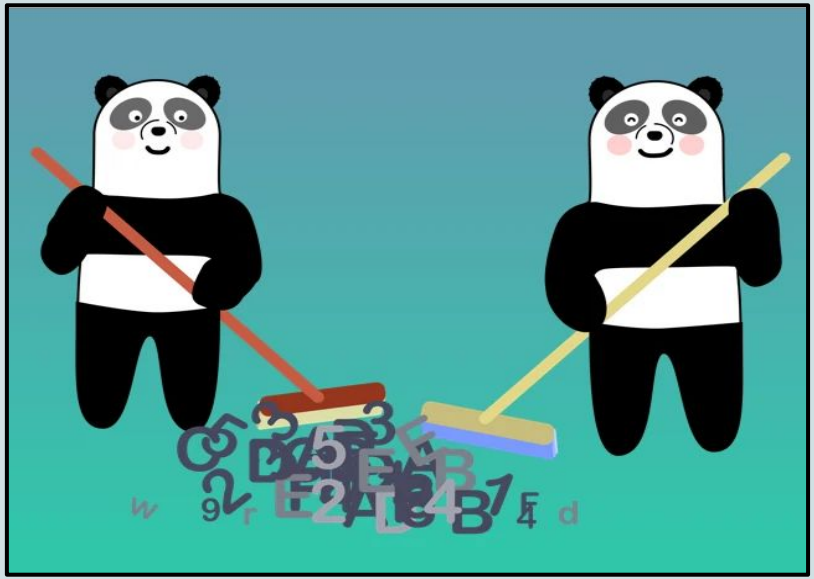
- Kaggle data set of approximately 20,000 chess games
- Examples of columns:
  - *‘winner’, ‘victory status’, ‘white rating’, ‘black rating’, ‘opening name’*
- **Has column of all moves in a chess game in PGN format**



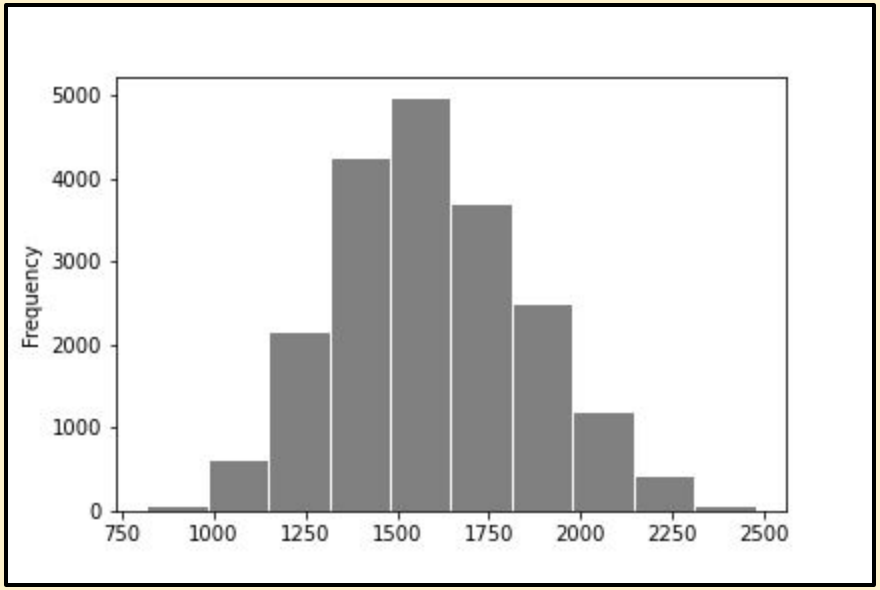
Previewing a PGN file

Data Cleaning

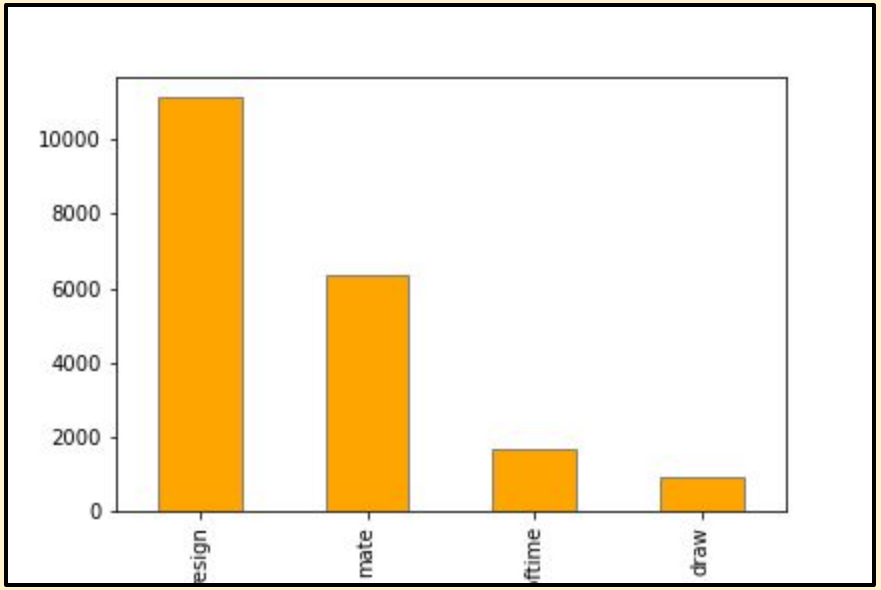
- Dropped a few unnecessary columns
- Split the move column into lists for modeling preparation
- Created new column: ‘average_rating’ for EDA



EDA (I): Average Rating & Game Outcomes



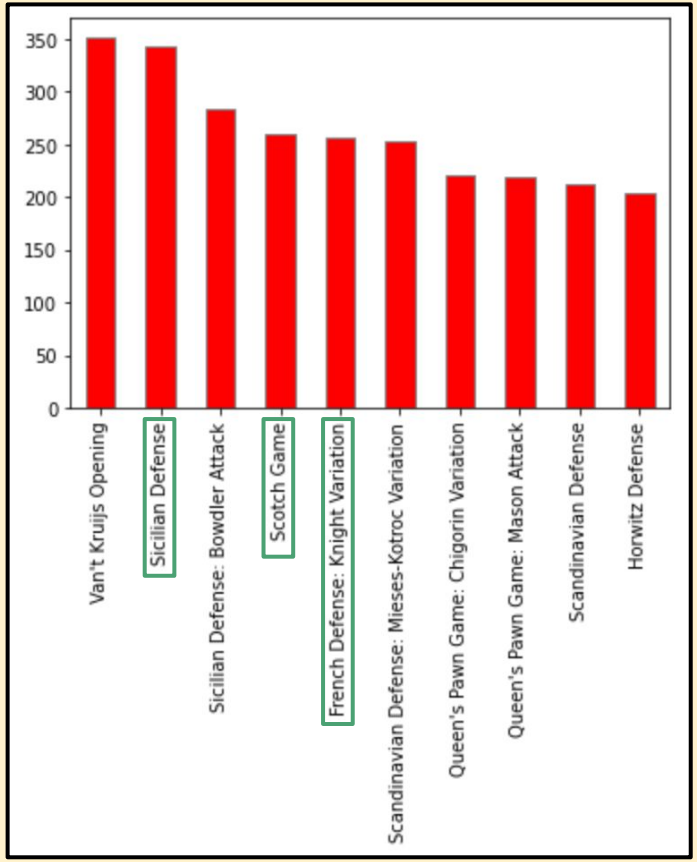
Ratings



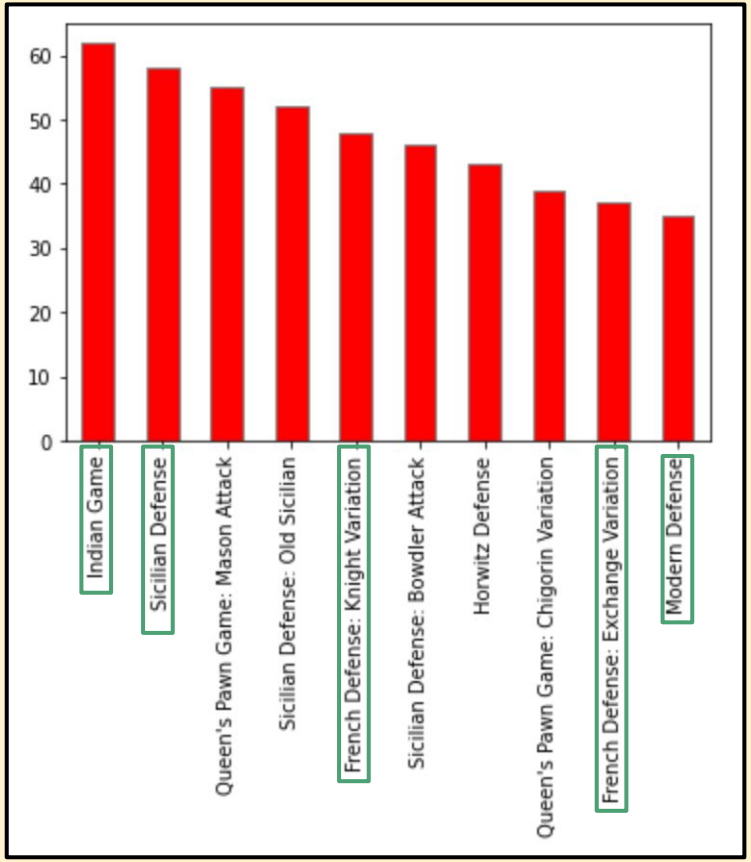
Outcomes (Resign, Checkmate, Out of Time, Draw)

EDA (II): Chess Opening Usage

All Ratings

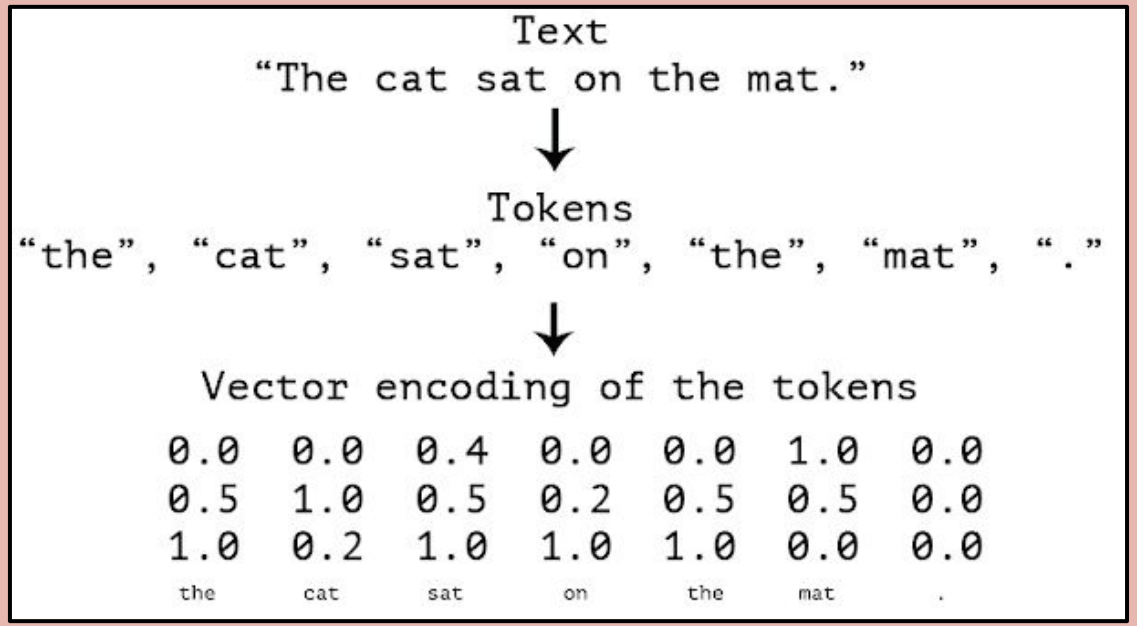


1700+ (Very Skilled)



Preprocessing - How can a large string of moves be predicted?

- Dropped all “draw” outcomes
- Made target variable and split lists of moves into an array of lists
- Keras’ Tokenizer
  - Padded sequences



Modeling (I) - LSTM vs GRU

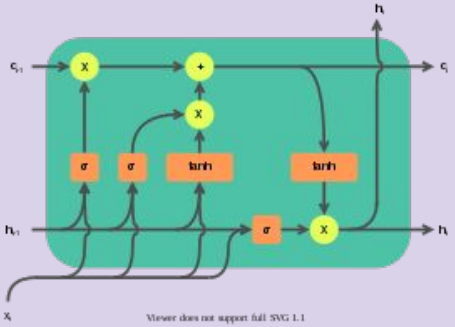
These two models are closely related

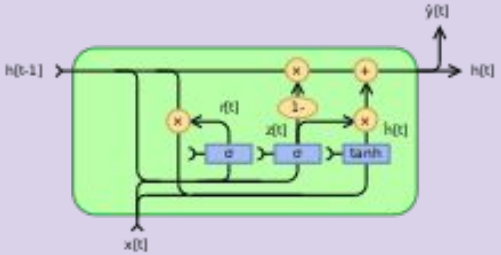
LSTM:

- Has been around for longer
- 3 gates (input, output, and forget gates)
- Uses memory unit
- Less computationally efficient (but. . .)
- Remembers longer sequences

GRU:

- Much newer Neural Net mechanism
- 2 gates (reset and update)
- Doesn’t use memory unit
- More computationally efficient
- Performs better on less training data





Modeling (II) - Results

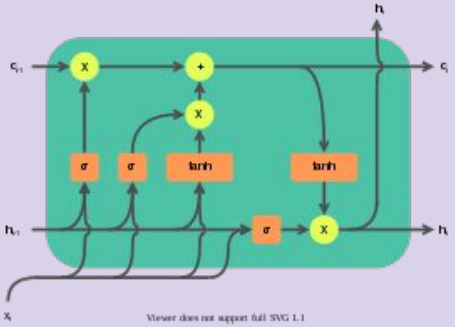
The GRU Performed Better than the LSTM

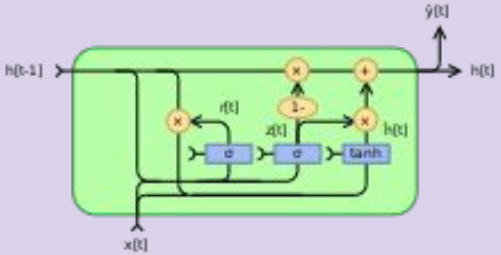
LSTM (Evaluated on test data):

- **50%** Loss
- **80%** Accuracy
- **86%** AUC
- Tested over 8 Epochs

GRU (Evaluated on test data):

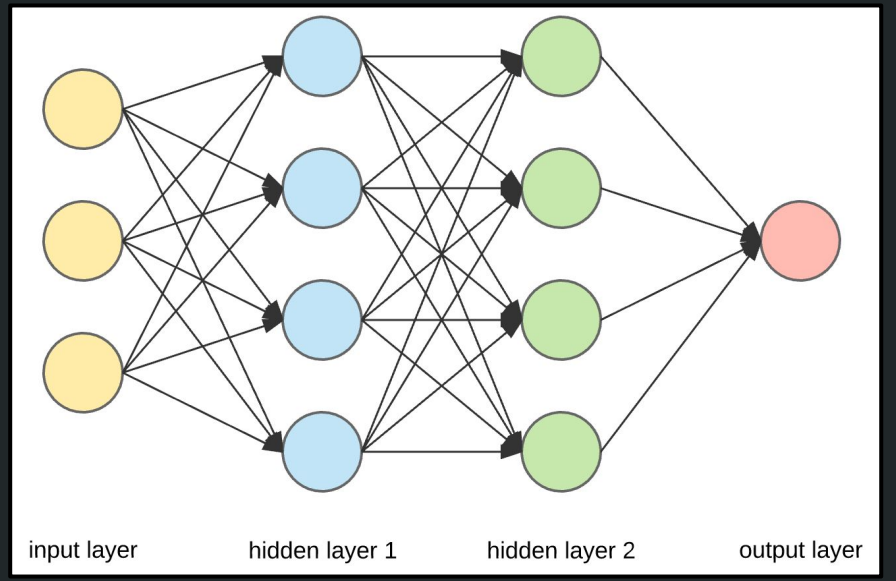
- **30%** Loss
- **88%** Accuracy
- **95%** AUC
- Tested over 7 Epochs





Modeling (III) - Limitations & Potential “Risks”

- Runtime
- Layer depth
- Optimization
- Wrong predictions
- Data set size



Conclusions & Recommendations

- Basic GRU neural network performs better and is more efficient than LSTM
- Chess move sequences are very deep and interrelated
- RNNs can be used for positional evaluation
- Predicting the outcome of a chess game only using the moves can lead to an excellent model

Moving Forward

- App on RNN for demo
- GridSearch on hyperparameters
- Fine-tune models
- Look into opening theory
- Look into other models
  - (Conv1D, Other CNNs)