

# Deep Convolutional Value Networks for Arimaa

Anonymous Author(s)

Affiliation

Address

email

## Abstract

Arimaa is an abstract strategy game with a small, but loyal community. Invented in 2002 by Omar Syed, the chess variant was designed to be intuitive for humans to learn, but difficult for computers to beat. Syed challenged the AI community to develop a World Champion program and further the state of knowledge about the game. The challenge has been met and finally claimed by David Wu in April of 2015. This paper increments on Wu’s results by substituting his hand-tuned evaluation function for a convolutional neural network, learned from a collection of expert games. Given positional features as input, the network will output the goodness score for one side. It hopes to serve as a proof-of-concept and eventually further the state-of-art in Arimaa playing strength.

## 1 Data Sets

Arimaa.com game room [1] offers an archive [2] of all games played on their servers. These include casual, rated, and tournament games for all users monthly from 2002-2016. I will use a subset of this data (minimum ELO rating 1800 and above) amounting to tens of thousands of game positions. Arimaa.com also offers a dataset of ‘win in two move’ puzzles [3], and I propose a method for extracting more. Data is in tab-delineated CSV, and game data is in move list form. These games require preprocessing to extract the key positions from them.

## 2 Problem

Can we train a convolutional neural network to learn a function  $f : \mathbb{R}^D \rightarrow \mathbb{R}$  to determine the goodness of a position for one side? How reliably do CNN evaluators determine the outcome of an Arimaa game? These are the machine learning inquiries I plan to address.

## 3 Methods

The data will need to be preprocessed, since it is stored as a list of moves. I will do this by extracting key positions in FEN notation [4], as the sole features, and game result (0 or 1) as the target.

RRRDDRRR/RHCEMCHR/8/8/8/8/rhcmechr/rrrddrrr 2g

Figure 1: An example FEN position string.

Parallel data preprocessing will be implemented via Apache Spark [5]. I will then fit a multi-layer convolutional neural network regressor in Python, making the proper transformations into the desired input shape. I will be using the software libraries Theano [6], and nolearn.lasagne [7] to design and train the network.

054  
055  
056  
057  
058  
059  
060  
061  
062  
063  
064  
065  
066  
067  
068  
069  
070  
071  
072  
073  
074  
075  
076  
077  
078  
079  
080  
081  
082  
083  
084  
085  
086  
087  
088  
089  
090  
091  
092  
093  
094  
095  
096  
097  
098  
099  
100  
101  
102  
103  
104  
105  
106  
107

## 4 Experiments

I will evaluate my model using  $k$ -fold cross-validation of mean accuracy values. I will break down accuracy results into move-by-move statistics. That is, I will plot my model on sample positions achieved after  $k$  moves,  $k + 1$  moves, ( $k + 2$  moves, etc). This will allow me to vet and visualize my model for stage-wise performance (i.e. opening, middle-game, end-game). I will produce a plot showing how well my model can predict the ‘win in 2,’ and ‘win in  $k$ ’ puzzles.

## 5 Related Work

Related work includes David Wu’s original paper [8], which I cite for the description of Arimaa, and his work developing a World Champion Arimaa program. I will compare my results to an earlier thesis paper by Hrebejk [9] on evaluator learning in Arimaa as a baseline for comparison. Oshri, Khandwala [10] is also relevant for its board to input image transformation for the CNN.

## References

- [1] Arimaa Gameroom; <http://arimaa.com>; 2002-2016.
- [2] Arimaa Games Archive; <http://arimaa.com/arimaa>; 2002-2016.
- [3] Arimaa ‘Win in 2’ puzzles; <http://arimaa.com/arimaa>; 2002-2016.
- [4] Forsyth–Edwards Notation; <https://en.wikipedia.org/wiki/>; 2016.
- [5] Apache Spark; <https://spark.apache.org>; 2016.
- [6] Theano; <http://deeplearning.net/software/theano/>; 2016.
- [7] Nolearn.lasagne; <https://pythonhosted.org/nolearn/lasagne.html> 2016.
- [8] Wu, David J. (March 2015). Designing a Winning Arimaa Program. ICGA Journal.
- [9] Hrebejk, T. (2013). Arimaa challenge - static evaluation function. M.Sc. thesis, Charles University, Prague.
- [10] Oshri, Barak. Khandwala, Nishith. (2015). Predicting Moves in Chess using Convolutional Neural Networks. Stanford University.