**The Data**

* **Initially, we did not know if we would be able to find sufficient training data for our project, and therefore planned to use a genetic algorithm, in which AIs would play against each other, in order to train without needing data.**
* **However, we were able to find a large collection of records of games played between human players on the site aagenielsen.dk.**
* **The site contains archived records of many games in the Hnefatafl family, including the variant, Irish Brandubh, which we are using.**
* **We scraped the data from the website using python libraries urllib and BeautifulSoup (an html parsing library)**
* **The data consisted of a list of moves by each player, which were isolated using regular expressions.**
* **The moves extracted could then be sequentially applied to the starting board state, producing the series of board states that took place during the game.**
* **We represented the board states as 3 lists of length 49, containing 1’s and 0’s indicating the absence or presence of a black piece, white piece, or king on a specific square of the 7x7 board.**
* **These 147 values are the inputs of the neural network, whose task it is to evaluate how advantageous the board state is for either black or white.**
* **The biggest issue relating to the data was how to assign a value to each board state for training and testing the neural network (this is the “target” value to which the output of the network is compared in the error function)**
* **To start, we assigned each board state a value of 1 or -1 based on whether black or white ended up winning the game, or 0 for a draw.**
* **Some timed games in the data set ended with a player timing out. These records were excluded from the training data, as the outcome is not necessarily correlated with which player was in an advantageous position.**
* **Next, values for board states which were repeated in the data set were averaged over all instances, giving a sort of probability of which player is most likely to win from that state.**
* **One important advancement we made was to add all possible reflections and rotations of each board state (8 in total, including the original) to the data set.**
* **This effectively increased the size of the training data set by a factor of 8, as well as theoretically allowing the network to be better at handling board states which are functionally identical, but vary due to being reflected or rotated.**