# Reinforcement Learning for Hanabi (Project Milestone)

**Vinson Luo** [1] **Arthur Tsang** [1] **Christopher Yeh** [1]

## Abstract

Hanabi is a collaborative card game where players can only see other players' cards and must rely on limited hints to share information. In this project, we approach Hanabi using the reinforcement learning technique of deep Q-networks (DQNs). We have implemented a simulator and feature representation of states, with preliminary results in progress. In the future, we'd like to add more subtle features to our representations as well as experiment with alternative approaches such as RNNs that can absorb information turn by turn.

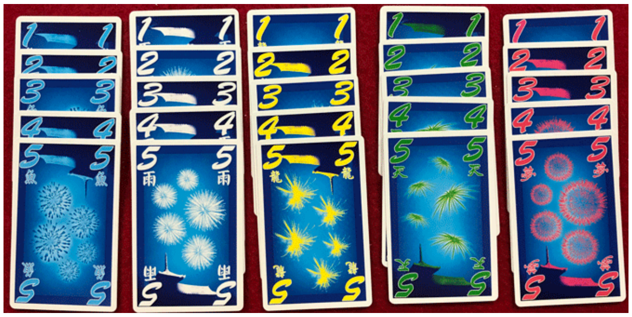

*Figure 1.* The Hanabi deck consists of cards numbered 1-5 in 5 different suits, as shown above. The arrangement shown above would give a perfect score of 25.

## 1. Introduction

Hanabi is a cooperative card game where all players can see the other players' hands, but not their own. Instead, one must infer about one's hand through specific hints from others.

In particular, Hanabi is played by 2-5 players using a specialized deck. The goal of the game is to play cards numbered 1 to 5 in sequential order in 5 different suits (as shown in Figure 1), with the score for all players at the end of a game being the number of cards successfully played, for a maximum of 25. It is noted that some luck is involved – certain initial shuffles of the deck can result in games where a perfect score is unattainable. Each player is dealt a hand of 4-5 cards, but players can only see the hands of other players, not their own. Players take turns taking one of three actions: they can either give a hint to another player (naming all the cards of a given color or number in the other player's hand), play a card from their own hand, or discard a card from their own hand. The players as a whole start with 8 communal hint tokens, which are expended as hints are given and can only be replenished by discarding cards or playing 5s. Additionally, if a player attempts to play a card which is not actually playable, the play counts as a mistake, and the game ends prematurely after the third mistake.

We argue that Hanabi is a particularly interesting arena to develop a cooperative AI agent. First, optimal play requires a theory of mind: since hints are limited, one must understand not only the information conveyed, but why a hint was considered by the other player to be the most informative or useful choice possible at a given point in time. Indeed, this game has been suggested as a measure of social intelligence (Osawa, 2015). Secondly, while many other games like Catan allow a subtle range of natural language communication, in Hanabi, every action, including the range of possible hints, is perfectly well-defined. Lastly, the game is not "too easy" in that a brute force approach would not work: not only is there an exponentially large number of possible hands and hints to remember, but even the ordering of hints can contain vital information.

## 2. Approach

Given the success of reinforcement learning in other games (Mnih et al., 2015; Silver et al., 2017), we approach Hanabi as a reinforcement learning problem, and in particular, at least to start, with a deep Q-network (DQN). Since Hanabi is cooperative, we can optimize a single policy for all players without a separate model for adversarial play. Existing literature has mostly focused on specific information-theoretic

[1] Department of Computer Science, Stanford, California, USA. Correspondence to: Vinson Luo <vluo@stanford.edu>, Arthur Tsang <atsang@stanford.edu>, Christopher Yeh <chrisyeh@stanford.edu>.

Copyright 2018 by the author(s).

analyses of the game. Cox et al., for example, developed the hat protocol, a hint protocol that encodes recommended actions or other information beyond the information in the hint itself (Cox et al., 2015). Baffier et al. have shown that determining whether or not a shuffled Hanabi deck can lead to a perfect score is an NP-complete problem (Baffier et al., 2016). By using reinforcement learning, we hope to create a model that will generalize well to other variants of the game, or perhaps one which also plays well with humans. We will be able to evaluate the progress of our AI by measuring the average score obtained by a simulator we built.

The game poses many levels of interesting strategy, with multiple possible state space representations. The size and presence of interactions in even a basic state space (which could be comprised of the identity of discarded cards as well as the history of hints) is large enough to warrant the use of deep learning. Later, we will experiment representing states with an RNN-type network which can read the history of actions at every turn. But to get started with something easier to train, we will treat each input state as a feature vector.

Our baseline feature space includes the following features:

1. Other players' cards and hints:
   - For each card, a one-hot vector of colors and numbers (length 10).
   - For each card, a one-hot vector of if a color or number hint was given (length 2).
2. Own cards and hints:
   - For each card, a one-hot vector of which colors and numbers are still possible (length 10).
   - For each card, a one-hot vector of if a color or number hint was given (length 2).
3. Number of communal hint tokens left, number of mistakes left.
4. Cards which have been successfully played:
   - For each suit, a one-hot vector indicating how many cards have been played so far (length 5, using **0** if no cards have been played yet).

In addition, we tag each card using a high-level notion very helpful to humans, namely whether a card is playable, indispensable, or dead (Cox et al., 2015). These are defined as follows:

**Playable** means that a card is valid to be played immediately without counting as a mistake.

**Indispensable** means that a card is the last remaining of its suit and number and must eventually be played in order to achieve a perfect score. Even without counting cards, since there is only one 5-card in each suit, one can tell a priori that these are indispensable.

**Dead** means that a card of the same type has already been played, so it will never be playable and is perfectly safe to discard.

Of course, a card is not always in one of the three states, and even if it is, it may not be apparent from a given player's point of view, but these notions give a sense of the kind of information that makes a good hint.

We have already implemented and begun to test the above feature space, and while it includes the most intuitively important information, there are still some relevant features that it leaves out. We would like to test a more advanced feature set which includes the features enumerated above, as well as:

5. Cards remaining in deck:
   - A count for each of the 25 types of cards remaining, from the player's point of view.
6. When each hint was given:
   - This information proves crucial for more advanced strategies in human play. Due to the constraints of the DQN architecture, it would have a finite length set by a hyperparameter, which represents a horizon beyond which our model no longer remembers when exactly a hint was given.

In addition, we implemented a replay buffer to better decorrelate $(s, a, s')$ triplets. That said, care must be taken to limit the staleness that comes from the fact that the same state implies vastly different information as a policy changes.

Having recently implemented a Hanabi simulator and baseline feature space, as well as a DQN model to train on it, we are in theory ready to present preliminary results. However, our low scores around 1 to 2 indicate that we may need to look harder for bugs or small design flaws.

## References

Baffier, Jean-Francois, Chiu, Man-Kwun, Diez, Yago, Korman, Matias, and Mitsou, Valia. Hanabi is np-complete, even for cheaters who look at their cards. 2016. URL http://eprints.sztaki.hu/9037/1/Baffier_4_3173858_ny.pdf.

Cox, Christopher, De Silva, Jessica, Deorsey, Philip, Kenter, Franklin HJ, Retter, Troy, and Tobin, Josh. How to make the perfect fireworks display: Two strategies for hanabi. *Mathematics Magazine*, 88(5):323–336, 2015. URL https://www.tandfonline.com/doi/pdf/10.4169/math.mag.88.5.323.

Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Rusu, Andrei A, Veness, Joel, Bellemare, Marc G, Graves, Alex, Riedmiller, Martin, Fidjeland, Andreas K, Ostrovski, Georg, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Osawa, Hirotaka. Solving hanabi: Estimating hands by opponent's actions in cooperative game with incomplete information. In *AAAI workshop: Computer Poker and Imperfect Information*, pp. 37–43, 2015. URL https://www.aaai.org/ocs/index.php/WS/AAAIW15/paper/view/10167/10193.

Silver, David, Schrittwieser, Julian, Simonyan, Karen, Antonoglou, Ioannis, Huang, Aja, Guez, Arthur, Hubert, Thomas, Baker, Lucas, Lai, Matthew, Bolton, Adrian, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.