

# Using Deep Q-Learning to Compare Strategy Ladders of Yahtzee

Philip Vasseur and James Glenn

## 1 Introduction

Yahtzee is a dice game that can be played with one (solitaire) or two players. We will be using Deep Q-Learning to calculate strategy ladders of Yahtzee for various sets of rules. A strategy ladder is a way of looking at how the performance of an AI varies with the computational resources it uses - in this project computational resources will be simply defined as training time for the reinforcement learning. Different sets of rules will change how the quickly the AI learns, and one can then compare the various strategy ladders to determine which set of rules gives the "best" strategy ladder shape. A "good" strategy ladder would be one that isn't too vertical or too horizontal, and rather gives a good combination of learning over time - it's important to note that this is all relative to the game itself. We assume/expect that there is some correlation between strategy ladders for AI and strategy ladders for human, meaning that a game with a "good" strategy ladder for an AI indicates that game is interesting and challenging for humans.

## 2 Rules of Yahtzee

There exist 13 rounds in Yahtzee, where the player(s) are awarded some points based on what dice they roll and what category they choose. In solitaire Yahtzee, the goal is simply to maximize your points. In two player Yahtzee, the goal is to get more points than your opponent.

In each round player(s) roll 5 dice. They then have the option of rerolling some subset of the dice up to three times (for each round). At the end of a round, they must choose a category to use, earning them a certain number of points depending on which combination of dice have and which category they chose. Generally, each category can only be used once. Categories are split between the upper section and the lower section. In the upper section the categories are aces, twos, threes, fours, fives, and sixes - where the number of points awarded is simply the sum of the dice with that categories number on them. In the lower section, the categories are three of a kind, four of a kind, full house, small straight, large straight, chance, and Yahtzee. The scores of these vary between the sum of the dice and a set number of points, depending on the category.

### 2.1 Bonuses

Especially in solitaire Yahtzee, high scoring strategies tend to be put a lot of importance on the two bonuses available to players. The first is the upper section bonus. The upper section bonus (in the official rules) is a bonus of 35 points awarded to a player if they score more than 63 points total in the upper section. The second is the Yahtzee bonus. If a player has already been awarded 50 points for using the Yahtzee category, a subsequent Yahtzee earns them another 100 points. Parameters such as

these are seemingly somewhat arbitrarily chosen, and as such are good examples of values we will vary when comparing strategy ladders.

## 2.2 Solitaire vs Two Player Yahtzee

The main difference between solitaire and two player Yahtzee is that in two player Yahtzee the goal is to get more points than the other player, rather than just maximizing your own points. Further, while solitaire Yahtzee has a state space small enough for an optimal solution to be found, this is not the case with two player Yahtzee which approximately squares the size of the state space. Regardless, solitaire and two player Yahtzee are still very similar, but there is a clear distinction between their optimal strategies. A risk-vs-reward element exists where going for a certain category-dice combination might be the best in terms of expected value of points, but in two player Yahtzee might not be the most optimal move. Consider a scenario where an agent has gotten unlucky and is very behind their opponent in points. Here, the move with the highest expected value of points might be a something very low risk though with a low reward. However, you could be effectively guaranteed to lose with such a low scoring move if you were, for example, in the final round. However, a much higher risk but higher reward move might have a worse expected value in terms of points, but be your only chance at winning. Considering these differences, it seems intuitive that an optimal solitaire Yahtzee player would surely do well at two player Yahtzee, but there is clearly room for improvement.

## 3 Project Goals and Information

The main goal of this project will be to use Deep-Q Learning to create a near optimal solitaire Yahtzee AI and then train this AI while changing various rule sets in order to compare the strategy ladders. The second goal will be to take this solitaire Yahtzee AI and train it with self-play and Deep-Q Learning to obtain the same results as above, but for two player Yahtzee.

### 3.1 Why Reinforcement Learning

While solitaire Yahtzee can be optimally solved, we will be using reinforcement learning in this project for two main reasons. First, the comparison of strategy ladders for solitaire Yahtzee has been done by a previous graduate student using supervised learning and defining the computational resources as the size of the neural network. Using Deep-Q Learning will be interesting for solitaire Yahtzee as it will allow us to see if the same results track over different definitions of computational resources. Second, we also hope to apply these same results to two player Yahtzee, a game whose state space is simply too large to use supervised learning - making reinforcement learning a much more reasonable choice.

### 3.2 Deep-Q Learning and Self-Play

After researching various libraries for Deep-Q learning and self-play, I've come to the conclusion that most of the work will have to be done by hand. For Deep-Q Learning, the best out of the box library

seems to be framework called Huskarl, but it's very new and not incredibly flexible. It will be good to build a base off of, as it has implementations of Deep-Q Learning and other adaptations to it that help convergence (Double Deep-Q Network, Dueling DQN). However, for example, the APIs to train and test are simply very limited and don't allow for much flexibility. Further, the framework seemingly does not work easily in terms of self-play where you have to make changes to your environment mid-training. Due to these limitations, I will likely fork the framework and only use the more raw functionality they've implemented, adapting it where necessary for my specifications. In terms of self-play, there does not seem to be any clean, out of the box libraries. Therefore, I will implement it myself and the solitaire Yahtzee AI as a basis agent - having it play against itself with a new objection of simply getting more points than its opponent. Using Deep-Q learning to have the agent gradually improve, the opponent will then be replaced with the better version.

### 3.3 OpenAI Gym

OpenAI Gym is well know framework for creating environments to test reinforcement learning algorithms. Husarkl uses OpenAI Gym environments, though as we won't be using Huskarl out of the box, it isn't necessary to use OpenAI Gym for this purpose. Regardless, OpenAI Gym creates a clean and clear API for usage within reinforcement learning problems. Using this API will allow us to easily test out algorithms on a number of other simple, pre-made environments. This is useful as it allows us to debug on less complicated environments - making sure our reinforcement learning algorithms work on simple, previously solved problems. This will allow us to more easily isolate any problems we run into during development.

## 4 Deliverables

- **Target: 9/19/2019** - Submit project proposal
- **Target: 10/04/2019** - Adapt Yahtzee game code into OpenAI Gym environment
- **Target: 10/23/2019** - Develop DQN agent for solitaire Yahtzee
- **Target: 11/01/2019** - Compare strategy ladders of solitaire agent over various rules
- **Target: 11/08/2019** - Modify openAI environment to work for two player Yahtzee
- **Target: 11/29/2019** - Implement self-play and develop DQN agent for two player Yahtzee
- **Target: 12/06/2019** - Compare strategy ladders of two player agent
- **Target: 12/13/2019** - Submit final report