

Reinforcement Learning Agent to Solve 2048 Effectively

Robrecht Conjaerts & Youri Coppens & Ruben Vereecken
Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium

May 28, 2015

Abstract

Samenvatting van de paper

1 Introduction

Since the explosive growth of smartphone usage in our everyday lives, and the improvement of SDK's for smartphone development, we have seen a rapid evolution from software developers to mobile developers. This led to a huge increase of applications for smartphones, with over 1.43 million applications for the Google Play store, and 1.21 million for the Apple App store as of December 2014 [1]. One of the more popular applications of 2014, was the puzzle game "2048" [2]. It is the objective of the player to increase its score by sliding the board (a grid 4x4) such that the same numbers that are next to each other merge, and multiply. These numbers are all powers of 2, and the higher the number gets, the higher your score will be. Figure 1 shows what the game looks like after the player has performed a couple of moves. We created an artificial intelligence (AI) that plays the game, and aims to gather a score as high as possible. Using reinforcement learning (RL) [4] we train our agent to learn how to play the game, and how to differentiate between good, and bad actions. This is done by using tile coding [4] to create a feature vector for the state the agent finds itself in, and Q-Learning [4] to update its Q-values. We start the paper by giving background information about the used techniques, and their functionalities. We will then explain in-depth how these methods are integrated in our algorithm, and how they help to learn how to play 2048. We show the results we managed to achieve, and examine them in detail. The paper ends by discussing future possibilities to achieve better results.

2 Background Information

This section was created to explain the basics of the techniques we will be using in our algorithm, to help our agent learn how to play the game 2048.

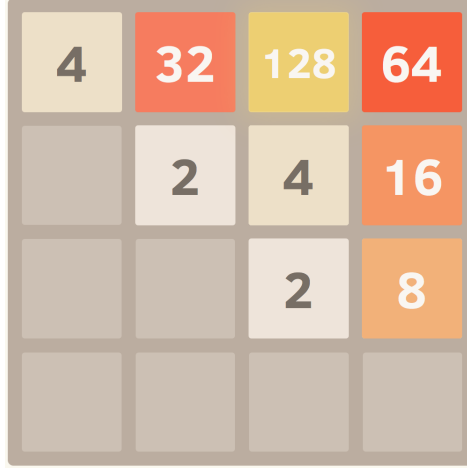


Figure 1: The board of 2048, after a couple of moves

2.1 Reinforcement Learning

When an agent interacts with an environment, and it receives a reward for this action, he will be able to distinguish between good, and bad actions based on this reward. This is what we call reinforcement learning [4], which we can define as a Markov decision process [3], that is a 5-tuple (S, A, P, R, γ) where:

- S is a finite set that contains the states of the environment
- A is a continuous set with actions available in each state
- P is the probability distribution that assigns a probability to each state, action pair $(S \times A) \mapsto [0, 1]$
- R contains the expected immediate rewards (given) for performing an action in a state $(S \times A) \mapsto \mathbb{R}$
- γ is the discount factor

From the environment an agent will perceive the state $s_i \in S$ he is in. Bounded to this state a set of actions $A_i \subset A$ that the agent can choose to perform. The agent has a policy π that decides which action the agent is going to take. For each state-action pair (s_i, a_j) the agent will receive a reward r , and uses this reward to update its policy, such that it will learn how to maximize the cumulative sum of rewards. Thus if we would perform an action a_j , in a state s_i , and would be returned a positive reward r the probability of choosing that action a_j would increase. The agent does not need to know beforehand which actions are good, and which actions are bad, he learns this by interacting with the environment, and it is in there that the power of RL lies.

2.1.1 Q-Learning

One of the best-known algorithms to solve a RL problem is the Q-learning algorithm [4].

2.1.2 Generalization and Function Approximation

3 Methods

In detail de methodes bespreken die we gebruiken, de functionaliteiten, eigenschappen en zo

4 Experimental setup

Het probleem 2048 goed uitleggen, en als we parameters in onze functies hebben, zeggen waarom die gekozen en zo.

5 Results

Resultaten tonen, en bespreken

6 Discussion and Future work

Praten over het nut van dit, wat het opgeleverd heeft, en hoe we het kunnen verbeteren in de toekomst

7 Acknowledgement

References

- [1] Ariel. App stores growth accelerates in 2014. Webpage, January 2015.
- [2] digiplex.in. 2048. Digital, December 2014.
- [3] Ronald A. Howard. *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, Massachusetts, 1960.
- [4] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.