Chapter 10
Game Playing (2048) Using Deep Neural Networks

Narendra Kumar Rao Bangole
Mohan Babu University, India

R. B. Moulya
MB University, India

R. Pranthi
Mohan Babu University, India

Sreelekha Reddy
Mohan Babu University, India

R. Namratha
Mohan Babu University, India

ABSTRACT

A one-player game, 2048 is also known as stochastic puzzle. This fascinating and engaging game has gained widespread acclaim and drawn researchers to create gaming software. The game 2048 has evolved into an engaging and difficult platform for assessing the efficacy of machine learning techniques because of its simplicity and complexity. Convolutional neural networks were used to create some computer players, but they performed poorly. In this work, the authors create a 2048 agent based on the reinforcement learning method and neural networks. The authors want to outperform other neural-network-based competitors in terms of results. Additionally, cutting-edge software created using this methodology for 2048 achieves the best performance out of all learning-based algorithms.

DOI: 10.4018/978-1-6684-9809-5.ch010

INTRODUCTION

A 4 × 4 grid is used for the game 2048. It is simple to learn yet challenging to master. To get to a 2048 tile in the original Game 2048, move and merge the tiles on the board in accordance with the regulations. In the initial state, two tiles bearing the numbers 2 or 4 are randomly arranged. The player chooses one of four directions—up, right, down, or left—and all the tiles travel in that direction after that. A tile with the sum value is created when two identically numbered tiles collide, and the player receives the sum as their score. Newly formed tiles do not merge once more in the same move since the merger happens from the opposite side. Additionally, the player is not able to choose a route in which no tiles merge or travel. The game finishes when the player is not able to move any more tiles, which happens when a new tile arrives at random in a cell with the numbers 2 or 4.

In the 2048 game, a straightforward puzzle game, the player joins tiles with the same number to produce a tile with a greater number. Making the appropriate actions at the right moments will help you get to the 2048 tile, which is the game's goal. The software must include a reinforcement learning algorithm, a neural network architecture to calculate the probable reward for performing a particular action in a specific state, and a game environment simulation or emulator to give the agent a way to interact with and learn from other players in the 2048 game. The software must also have a user interface, which can either be a simple API for automated interaction or GUI for human players.

The software can learn to play the 2048 game optimally, maximizing the game score and win rate, by training the agent via reinforcement learning.(Shilun Li et al.) The agent may be educated on a lot of data, which enables it to gain knowledge from a variety of situations and gaming scenarios. An agent created using this method may be more effective and efficient than one created using conventional game-playing algorithms. In this work, the objective is to develop an agent that can master the game on its own and get a high score. The agent will be trained using a reinforcement learning algorithm, where it will be rewarded for actions that increase its score and penalized for those that decrease it. Here the agent will be trained by watching how the game is played, making decisions based on its policy, and being rewarded for acts that increase the score. The agent will be able to learn intricate patterns in the game and adapt to various game situations by the usage of deep neural networks. The work not only sheds light on how deep reinforcement learning can be used to gaming, but it also advances the discipline by offering a fresh approach to the game 2048.By using reinforcement learning we can increase efficiency also when compared with convolutional neural networks. By using convolutional neural networks that performance might be slow and not efficient. So this deep reinforcement learning algorithm plays a major role while the agent is playing games. This reinforcement learning is a sort of machine learning that focuses on instructing an agent to make decisions by rewarding or penalizing its activities. Reinforcement learning algorithms are used in artificial intelligence for various purposes like game playing where this learning is used to train agents and play games like chess, etc., and also this is mainly used in robotics, in controlling robots, allowing them to learn how to perform tasks and navigate their environment.

This reinforcement learning is used in autonomous vehicles and is applied in developing self -driving cars, allowing them to make decisions on their own and navigate roads and avoid obstacles. In addition to operating complex systems like power grids and water distribution networks, this algorithm also aids in system performance optimization and the reduction of energy usage. In these ways, the reinforcement learning algorithm is used in various domains in artificial intelligence. (Nathaniel G et al., Sutton R. S)