# Convolution and Deep Learning in Strategy Games

Madalin-Augustin, Raiu

*2022-2023*

# Summary

## Clasification

ACM ( 1998 ): K.8.0 , F.1
AMS ( 2020): 42A85
Key Words: Gaming, AI in Strategy Games, Neural Network

# 1. Introduction

## 1.1 An Introduction to Convolutional Neural Networks

## 1.2 Understanding of Convolutional Neural Networks

## 1.3 Real-Time Strategy Games

## 1.4 Player – AI Interaction

# 2.Related Work

## 2.1 Convolution

A convolution is the simple application of a filter to an input that results in an activation. Repeated application of the same filter to an input results in a map of activations called a feature map, indicating the locations and strength of a detected feature in an input, such as an image.

## 2.2 Neural Network

When you ask your mobile assistant to perform a search for you—say, Google or Siri or Amazon Web—or use a self-driving car, these are all neural network-driven. Computer games also use neural networks on the back end, as part of the game system and how it adjusts to the players, and so do map applications, in processing map images and helping you find the quickest way to get to your destination.

A neural network is usually described as having different layers. The first layer is the input layer, it picks up the input signals and passes them to the next layer. The next layer does all kinds of calculations and feature extractions—it's called the hidden layer. Often, there will be more than one hidden layer. And finally, there's an output layer, which delivers the final result.

A neural network is a system or hardware that is designed to operate like a human brain.

## 2.3 Predicting human behavior in size-variant repeated games through deep convolutional neural networks

Predicting the behavior of individuals in strategic interactions has always been considered as an important task in economics, politics, psychology and social behavior analysis. In the last decades, numerous game-theoretic learning models have been proposed to model the players' bounded rationality observed in empirical interactions. These models aim to explain the players' deviations from Nash equilibrium, while they are tended to be as accurate as possible

## 2.4 Using a scouting Strategy for Real-Time Strategy Games

The primary goal of scouting strategies is to collect the spatial data in a certain area. It is similar to robotic exploration of unknown terrain. Collecting and analyzing terrain information is commonly perceived to be a key foundation in forming RTS gameplay strategies, due to the vital information it contains.

## 3.Method

One particular strength of convolutional neural networks is the ability to detect features. They perform well not only in identifying simple features such as edges but also combining these simple features to understands more complex characteristics of the game. In this article I'm trying to see the different methods you can implement neural networks in strategy games and how can you get it close to playing like a Human.

## 4.Conclusion

## 5. METHOD

### 1.Min Max Algorithm in AI

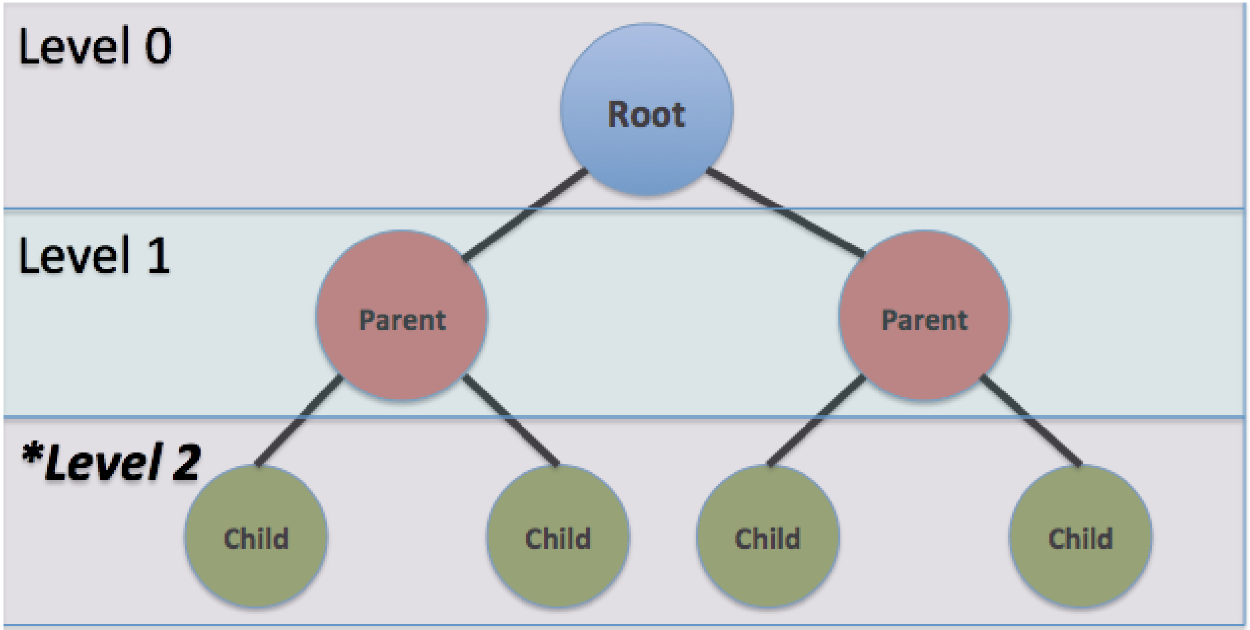
A first step in working with AI in strategy games is the min max algorithm.

The min max algorithm in AI, popularly known as the minimax, is a backtracking algorithm used in decision making, game theory and artificial intelligence (AI). It is used to find the optimal move for a player, assuming that the opponent is also playing optimally. Popular two-player computer or online games like Chess, Tic-Tac-Toe, Checkers, Go, etc. use this algorithm.

#### *How does it work?*

In the min max algorithm in AI, there are two players, Maximiser and Minimiser. Both these players play the game as one tries to get the highest score possible or the maximum benefit while the opponent tries to get the lowest score or the minimum benefit.

The goal is to find the best possible move for a player. This can be done by choosing the node with the best evaluation score. The best choice will be made after evaluating all the potential moves of the opponent. The algorithm looks ahead at all the possible values till the end and makes a decision for the player.



The game tree above is a nested data structure that is used to evaluate the moves. Here the root node is Level 0, which branches out into Level 1 or parent nodes, which further branch out into Level 2 or child nodes. The branching out can continue to many levels, having the potential of infinite levels. Level 0 is like the current state of the board, while Level 1 is all the possible states of boards depending on the next move.

### *The steps for the min max algorithm in AI can be stated as follows:*

- Create the entire game tree.
- Evaluate the scores for the leaf nodes based on the evaluation function.
- Backtrack from the leaf to the root nodes:
- For Maximizer, choose the node with the maximum score.
- For Minimizer, choose the node with the minimum score.
- At the root node, choose the node with the maximum value and select the respective move.

### *Properties of min max algorithm in AI*

- The algorithm is complete, meaning in a finite search tree, a solution will be certainly found.
- It is optimal if both the players are playing optimally.
- Due to Depth-First Search (DFS) for the game tree, the time complexity of the algorithm is O(bm), where b is the branching factor and m is the maximum depth of the tree.
- Like DFS, the space complexity of this algorithm is O(bm).

### *Advantages*

- A thorough assessment of the search space is performed.
- Decision making in AI is easily possible.
- New and smart machines are developed with this algorithm.

### *Limitations*

- Because of the huge branching factor, the process of reaching the goal is slower.
- Evaluation and search of all possible nodes and branches degrades the performance and efficiency of the engine.
- Both the players have too many choices to decide from.
- If there is a restriction of time and space, it is not possible to explore the entire tree.
- But with Alpha-Beta Pruning, the algorithm can be improved.

https://github.com/Raiu-Madalin-Augustin/research - a game i developed with min max algorithm in college 2 years ago

## 6.Bibliography

1. Generating Diverse and Competitive Play-Styles for Strategy Games

2. Understanding of a Convolutional Neural Network (https://www.researchgate.net/publication/319253577_Understanding_of_a_Convolutional_Neural_Network)
3. Recent advances in convolutional neural networks (https://www.sciencedirect.com/science/article/pii/S0031320317304120)
4. AI in Computer games: Smarter games are making for a better user experience.
5. A scouting Strategy for Real-Time Strategy Games (https://www.researchgate.net/publication/299441486_A_Scouting_Strategy_for_Real-Time_Strategy_Games)
6. An Introduction to Convolutional Neural Networks - Keiron O'Shea, Ryan Nash
7. Building Human-Level AI for Real-Time Strategy Games - Ben George Weber, Michael Mateas, Arnav Jhala
8. Real-Time Strategy Games: A New AI Research Challenge - Michael Buro
9. A multi-size convolution neural network for RTS games winner prediction - Jie Huanga and Weilong Yang
10. Evaluating real-time strategy game states using convolutional neural networks - Marius Stanescu; Nicolas A. Barriga; Andy Hess; Michael Buro
11. ELM-based convolutional neural networks making move prediction in Go - Xiangguo Zhao, Zhongyu Ma, Boyang Li, Zhen Zhang & Hengyu Liu
12. Predicting human behavior in size-variant repeated games through deep convolutional neural networks - Afrooz Vazifedan & Mohammad Izadi
13. Using Convolution and Deep Learning in Gomoku Game Artificial Intelligence - Peizhi Yan and Yi Feng
14. Convolutional and Recurrent Neural Network for Gomoku - Rongxiao Zhang
15. Player-AI Interaction: What Neural Network Games Reveal About AI as Play -