General Video Game AI (GVG-AI): Learning from screen capture

Kamolwan Kunanusont, Simon M. Lucas, Diego Perez-Liebana

Introduction

This paper explores the idea of general video game artificial intelligence using deep Q Reinforcement learning and convolutional neural networks. It proposes a single learning framework which can be used to play games of different categories and difficulty levels. The inspiration for this work is drawn from human beings who are capable of solving several kinds of problem efficiently and develop a general artificial intelligence agent (GVG-AI). It processes the information on the screen using Convolutional Neural Network (CNN) and determines the best move using Deep-Q-Network. General Video Game Playing (GVGP) agent takes action based on the encapsulated information from the game.

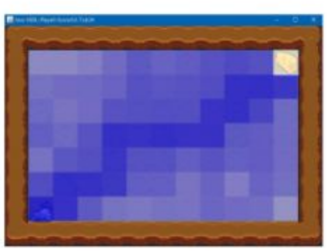
Proposed Method

Pre-processing unit: Two pre-processing schemes were proposed to support both visualize and non-visualize mode for GVG-AI framework. In order to support visualize mode, each screen size was transformed to fixed size based on the CNN architecture.Non-visualization pre-processing generates screen information from a framework-provided object called gridObservation. Each sprite type is mapped to a random RGB color, then an image is generated based on gridObservation which is normalized and expanded if necessary.

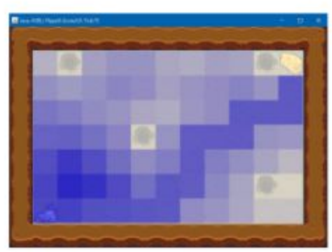
Learning Unit: Two CNN architecture were proposed to train the network with kernel size of wither (5x5) or (3x3) depending on the network architecture. There are 32 and 64 neurons in the first and second convolutional layers respectively. Stride size was chosen to be 1x1, subsampling kernel size is 3x3. Dense layer consists of 512 neurons and output size is same as number of available actions for the game. The agent performs learning update in three occasions - during experience creation, sampling and at the end of the episode.

Experimental results:

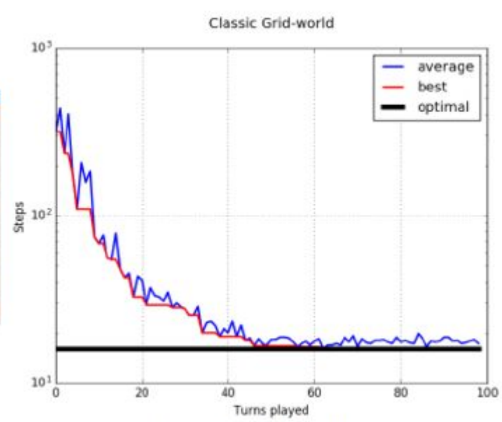
The figure below on the left shows the heat map indicating how frequently the agent visited each position while playing the game of grid-world. Number of turns required to win the game is shown on the right It is clear that the the algorithm took fewer steps to win the game as it played more and found the optimal (16) steps within 60 turns.



(a) Original Grid-world



(b) Trapped Grid-world



Classic Grid-world steps taken.

Conclusions

This paper is a step towards building a general artificial intelligent system which is an active area of research. The learning agent was able to solve both static and stochastic games. Some future work in this area could be to choose CNN network based on the game's complexity. Another area to explore could be using transfer learning by transferring the training information from one game playing agent to another akin to human intelligence.