Daniel Marsh
Advisor: Professor J. Glenn
CPSC Senior Project

Transfer Learning Within a Single Game

Introduction

Computational intelligence for games has been a major field of research in artificial intelligence since the creation of Nimatron in 1940. The field continued to advance over the next several decades and by 1989 a computer program known as Chinook was able to play expert-level checkers. Five years later in 1994, the Chinook program won the Man-Machine world championship in checkers. Three years after that IBM's Deep Blue machine beat the world's top ranked chess player. By 2016, Google's Alpha Go program surpassed expert human players in Go. Today, computational intelligence for games remains a significant area of research in artificial intelligence.

Motivation

I was first exposed to this field in CPSC 474, Computation Intelligence for Games with Professor Glenn, where I learned about and implemented several common computational intelligence algorithms in the context of games. After completing the course, I realized that I wanted to explore the field further and attempt to solve a more complex problem.

Many of the early successes in machine learning for games came from agents being trained from scratch on each game. This is obviously not how humans learn to play games as a good strategy in one game may inform a good strategy in another. Humans have the ability to transfer their learned strategies from one game to another rather than just learning each from scratch. The concept of transfer learning opens the door for a universal gameplaying agent or, more generally, machine learning applications that can transfer skills from one task to another. For my project, I will explore different machine learning techniques and their compatibility with transfer learning in the simple case of transferring knowledge between different versions of the game Can't Stop.

Project Overview

My project will center around the board game Can't Stop. Rules linked below: https://www.ultraboardgames.com/cant-stop/game-rules.php. Can't Stop is currently too computationally complex to solve using a simple search algorithm but has several aspects that can be simplified to make it solvable in a reasonable amount of time. I will exploit these aspects to create and solve a simplified version of Can't Stop. I will then use three training methods to train agents on the simplified solved version of the game. Finally, I will test the three trained agents on the original version of Can't Stop to evaluate how well the knowledge of the simplified game can be transferred to the full game. In addition to the three agents trained on the simplified game, I will implement several agents using simple computation intelligence methods to test against the transfer learning agents.

Project Details

As stated above, my project will implement three agents trained on the simplified game, as well as several agents using common computational intelligence methods. The simple agents will include a random agent, a minimax agent (depth 4), and an agent with simple game knowledge. These will initially be used to test the rule implementation and later to test the three more complex agents.

The first transfer learning agent will be a variation of a Markov decision-making algorithm that uses a superstate bucketing scheme to handle state inputs of variable length. I will use a dynamic programming approach to solve the expected value of each possible game state in my simplified version of the game. I will then populate the superstate bucketing scheme with these expected values. When playing the full game, this agent will translate a state into a superstate using the bucketing scheme and use the corresponding value as an estimated expected value for the given state. The agent will then choose the next action by using a Markov decision process and the estimated expected values. This approach can be enhanced by using the estimated expected values for the full game to initialize a Q-learning algorithm. The success of this system can be measured by testing it against a Q-learning agent that does not use any transfer learning, but trains from scratch on the full game.

The second transfer learning agent will use a Long Short-Term Memory (LSTM) network. LSTM networks are designed to handle problems with variable input size. For a game like Can't Stop, the state descriptions for the simplified game are smaller than those for the full

game. I will use the solvable simplified game to train the LSTM network and then test on the full game to evaluate how well the training in the simplified game can be transferred to the full game.

The final transfer learning agent will use an autoencoder. Autoencoders are generally used to learn representations of sets of data. For this project, I will train an autoencoder to use the same state representation for any version of Can't Stop. Once that training is complete, the network can be trained to recognize good positions using encodings of the simplified game and apply this knowledge to the positions in the full game that have similar encodings.

Once implemented, each agent will be tested against the simple agents described above, as well as the two other transfer learning agents. I will conclude the project by reviewing the success or shortcomings of each approach in a project report, ultimately trying to determine the compatibility of transfer learning with various machine learning techniques.

Deliverables

The project will include both code submissions and a final report. The code submissions will include Can't Stop game rules, three simple agents as described above, a Markov decision agent, including the Q-learning enhancement, a LSTM agent, and an autoencoder agent. The project will also include a final report of no less than eight pages that includes analysis of the success or failure of each approach as well as detailed reports on the win percentages of each agent against all other agents.

Schedule

02/26: Rules and simple agents

03/12: Markov Decision Agent with Q-Learning Enhancement

03/26: LSTM Agent

04/09: Autoencoder Agent

05/01: Review and Report

05/13: Final Submission