Transfer Learning with FlappyBird

Cedrick Argueta
Department of Computer Science
Stanford University
Stanford, CA 94305
cedrick@cs.stanford.edu

Austin Chow
Department of Computer Science
Stanford University
Stanford, CA 94305
archow@stanford.edu

Cristian Lomeli
Department of Computer Science
Stanford University
Stanford, CA 94305
clomeli@stanford.edu

Abstract

Reinforcement learning's growth in popularity in recent years is partly due to its ability to play some video games with a level of mastery that no human can reach. Transfer learning is popular in the field of deep learning, and using pre-trained models on certain tasks speeds up training time and increases performance significantly. In this project we aim to apply transfer learning to the popular video game *FlappyBird* and analyze its performance compared to traditional reinforcement learning algorithms.

1 Introduction and Scope

2 Approach

2.1 Method

The goal of our project is twofold: we aim to evaluate deep reinforcement learning algorithms on the FlappyBird game, and also experiment with transfer learning and what impact it makes on the training process.

2.2 Expected Behavior

Austin, give input output behavior

2.3 Infrastructure

The infrastructure for the game comes mostly from the PyGame Learning Environment and keras-rl packages. The PyGame Learning Environment provides a nicely wrapped implementation of FlappyBird, complete with sprites and the relevant game mechanics built in. Keras-rl provides a deep reinforcement learning framework that provides a simple interface for training agents. We take advantage of these two here, writing simple wrappers for the PLE game instance so that it is compatible with keras-rl.

The keras-rl package additionally provides an implementation of several algorithms that are applicable to FlappyBird, particularly Deep Q-networks and the SARSA algorithm.

Preprint. Work in progress.

2.4 Baseline and Oracle

Cristian, define baseline and oracle

3 Challenges