# Quickly Training MuZero to Play Centipede

Erek Cox – Zac Pear – Joe Skimmons

**Abstract**

The MuZero model has been proven to perform better on most Atari games better than previous learning agents. These results came about after many hours of training on expensive hardware and are difficult to replicate without these resources. We attempt to simplify the process for MuZero on the Atari game "Centipede," training with modest resources in a much shorter time frame.

## 1 Proposal

### 1.1 Problem Formulation and Goal

[1] includes MuZero's results after 12+ hours training on expensive hardware (board games:16 training TPUs, 1000 self-play TPUs; Atari games: 8 training TPUs, 32 self-play TPUs; See [1]'s Appendix G). We want to see if MuZero's performance is feasible on modest hardware: a single GPU on a Google Cloud instance. We aim to modify [2] to operate on Centipede instead of Cartpole, then down-sample the Centipede input data from [3] being fed into the model. Training on the modest setup in a reasonable time frame (ideally 1 hour), performance will then be evaluated to see whether the model can perform on a high level (or at least show signs of improvement).

### 1.2 Dataset and MuZero Algorithm

Comprised of a representation function, a dynamics function, and a prediction function, the MuZero algorithm makes a prediction for policy, value function, and immediate reward at each time step given past observations and future actions. The dynamics function of the algorithm replicates the actions of a regular MDP model, except the state is not a direct representation of the environment, but a hidden state of the model and serves the sole purpose of predicting the values of the next policy, value, and reward. In the planning nature of the algorithm, the model uses a Monte Carlo Tree Search to output intermediate rewards. The given Centipede gym, 'centipede-v0' [3], returns data on the current game environment (time, state, reward, etc.) that will be used to evaluate and train the model for improved game performance. The Centipede game data will be downscaled using scikit-image [4], allowing the model to operate on a less complex dataset.

### 1.3 Evaluation Criteria

We will be evaluating the MuZero model's performance when playing Centipede by comparing its scores against those of other agents such as humans and previous models, as seen in [1] on page 17 and 18.

## Previous Work and References


[1] Thomas Hubert Karen Simonyan Laurent Sifre Simon Schmitt Arthur Guez Edward Lockhart Demis Hassabis Thore Graepel Timothy Lillicrap David Silver Julian Schrittwieser, Ioannis Antonoglou. Mastering atari, go, chess and shogi by planning with a learned model. *arXiv*, 1911.08265v2, 2020.

[2] Johan Gras. Muzero. `https://github.com/johan-gras/MuZero`, 2019.

[3] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. `https://gym.openai.com`, 06 2016.

[4] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014.