**Project Title:** Video Gaming with Neural Networks

**Name(s):** Russell Bisker

**Project description:** Explore different deep learning techniques that can be used to train a model to play retro video games. I hope to compare several options, such as the Deep Q-Learning technique used in the 2013 *Playing Atari* paper versus Proximal Policy Optimization (PPO) and/or incorporating transformers rather than standard CNNs.

**Project goal:** Use modern deep learning techniques and programming frameworks to build one or more models capable of successfully playing multiple retro video games in real time. Ultimately, I hope to produce several working models and compare performance across at least several games.

**Project timeline:**

* Learn more about Deep Q-Learning and Transformers and possibly PPO
* Write a python implementation of one or more deep learning models that interact in real time with video game emulator
* Test and optimize the model(s) across several games
* Produce final results and writeup report

**Data:** data will be generated by automatically playing the game(s) repeatedly via an API interacting with a game emulator. I’ve identified 2 frameworks that could work for this purpose, either [The Arcade Learning Enviornment (ALE](https://github.com/Farama-Foundation/Arcade-Learning-Environment)) or [Gym Retro](https://github.com/openai/retro)

**Resources/Key references:**

[*Playing Atari with Deep Reinforcement Learning*](https://arxiv.org/abs/1312.5602)

[Gym Retro](https://openai.com/index/gym-retro/)

[ALE Documentation](https://ale.farama.org/)

[*The Arcade Learning Environment: An Evaluation Platform for General Agents*](https://jair.org/index.php/jair/article/view/10819)

[MarI/O (Youtube)](https://www.youtube.com/watch?v=qv6UVOQ0F44)

[Build a Mario AI model with Python (Youtube)](https://www.youtube.com/watch?v=2eeYqJ0uBKE)

[Proximal Policy Optimization](https://openai.com/index/openai-baselines-ppo/)

[Stable Baselines PPO](https://stable-baselines3.readthedocs.io/en/master/modules/ppo.html)