

Part A

My individual contribution to the project includes background research and understanding preliminary theory, testing the systems developed by my team on environments, and working with my team on ensuring the deep learning system was set up correctly for learning over time. This project has significantly improved my previous abilities in python development, machine learning, and deep learning with tensorflow, as all of our development was done using object-oriented Python while combining it with modern tensorflow best-practices used by others in our field. In addition, my research and personal experience in working with reinforcement learning and tensorflow have significantly improved my understanding of machine learning and deep learning and how these systems learn, as well as what sort of fine-tuning is required.

Object-oriented Python was certainly within my skill set prior to this project, but its interaction with tensorflow allowed for unique learning opportunities in regards to reducing redundancy and ensuring functionality was maintained across learners. As our project expanded and more learners were added, a generic learning class needed to be created to allow our team and others to utilize our code's basic testing functionality without duplicating it. This also required ensuring our systems worked with tensorflow and non-tensorflow learners, allowing for complete extensibility to any type of learner. Ensuring that multiple learners were able to all exist at the same time was another requirement which added complexity to the project with regards to Tensorflow, but I was successfully able to utilize the tensorflow graph system to ensure the separate networks didn't interfere with each other. Finally, ensuring that these multiple learners could all be saved and reloaded at regular intervals proved to be an interesting challenge, as Tensorflow's variable scope and graph systems are different from basic python serialization, but I managed to get it working and ensure that the system regularly saves the network to file in the event that the system crashes or the network needs to be loaded later. All of this gave me a strong understanding of Tensorflow's underlying systems which would be necessary for more complex deep learning problems which might not be solved in a matter of minutes or hours. One of my challenges was actually separate from deep learning itself, but rather installing NVIDIA drivers and deep learning tools on a Ubuntu laptop to ensure that the GPU optimizations were in place. Basic instructions provided by NVIDIA and Tensorflow for installing proprietary drivers proved to be somewhat unhelpful, as it caused the laptop to basically lose access to all graphical interactions. The entire install process set our project back a few days as I worked to get the system back to a working state with the drivers installed. Eventually I was able to get it working, but this type of example showed that project complications can arise from unlikely areas, even installing drivers.