For Reinforcement Learning models on Atari games in the past, most have not taken the approach of using image based models and they have received fair results after a very long training period, but with a image based model we can generate a high reward model in less than 100k straining steps which is equivalent to about 2 hours of game play. This model’s policy will be the Simulated Policy Learning (SimPLe) which build off of a Markov Decision Process. The architecture of the model consists of 3 sub models, the first of the models consist of a dense layer and multiple convolutional layers and this is to take in the 4 frame steps that the model will be working with as input. The next model will consist of three deconvolutional layers and at the end one final larger deconvolutional layer to output the predicted frame. Finally, the last model consist of a dense layer and two convolutional layers, an attention layer, and one last dense layer to predict the output.

In a different research paper we are looking at the process of training a reinforcement learning model to tackle the a more challenging problem, StarCraft II. The main long term goal of this project is to train a model where it’s good enough to consistently defeat top human players. Some challenges that comes with achieving this goal is that StarCraft II had a lot data that a model can take from it and reducing the data needed to only the important values and making it train in a reasonable amount of time is the main one, but this team of researchers have the support of Blizzard which is the studio that made StarCraft II and they are using an API to gather data as well as a computer vision system. StarCraft also has the unique challenge that is doesn’t allow you to see the whole map so an agent should constantly be looking to gather new data that it can use to make a valid prediction on what it’s next steps will be. Another aspect of this research is that in addition to making a model for the main gameplay, they are also making different models for minigames that appear throughout the game as well. Those will be trained differently and less complex than the main gameplay model. For the main game play they are using keyboard shortcuts to cut down the amount of action states there are and make it more manageable. The team is using the Win (1), Tie(0), and Loose(-1) with the numbers next to the events equaling the reward they would receive. On top of the end game result the team will also be using the Blizzard score value to cut down on the episode lengths and have a continuous reward system going. The main contributing inputs to are the StarCraft API provided by blizzard which allowed the research team to construct a python library that allows actions to be taken and information to be gathered through the API indirectly.