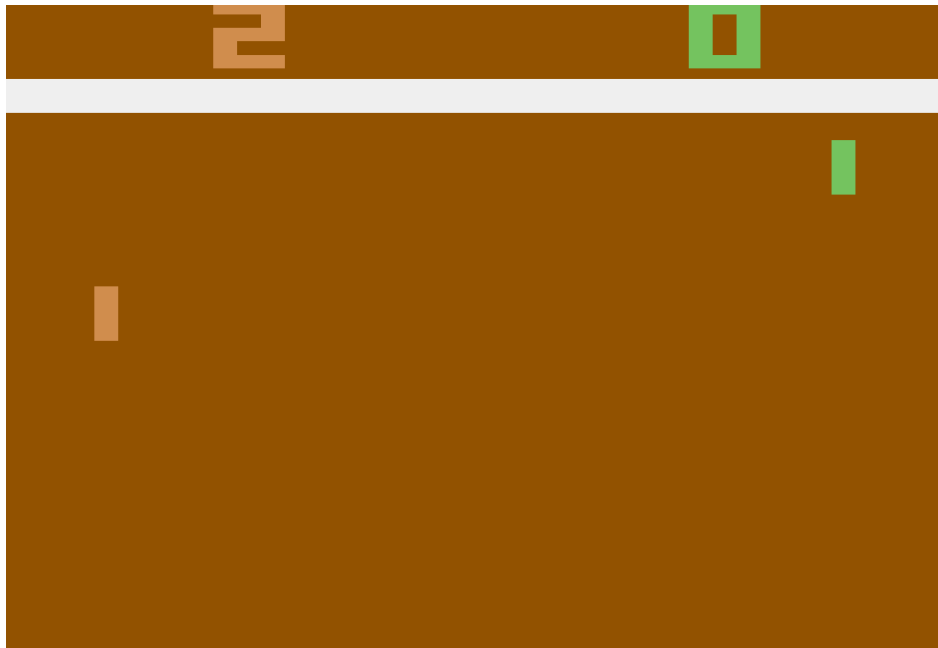


Final Project

For the final project, you will implement DQN using experience replay with prioritized replay, fixed Q-targets and double Q-Learning to solve the Atari Pong problem.

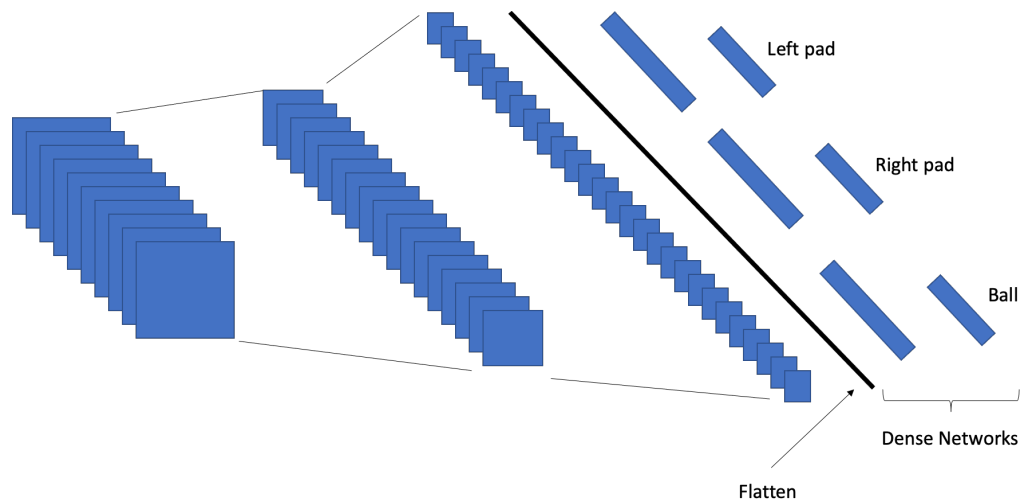
After training the network, you will present a short movie training the network, and play 100 games against OpenAI player. You will report.

Average number of steps until someone wins, and the score of the game. In case you need to train the model, I recommend you extract the positions and the images first and train the model in colab using a GPU. If you do not know how to do that, please contact me. Remember that GPU from Colab is free for 24 hours, so you need to constantly save the state of the keras or pytorch model and your training job so that you can resume if your Colab session is terminated.

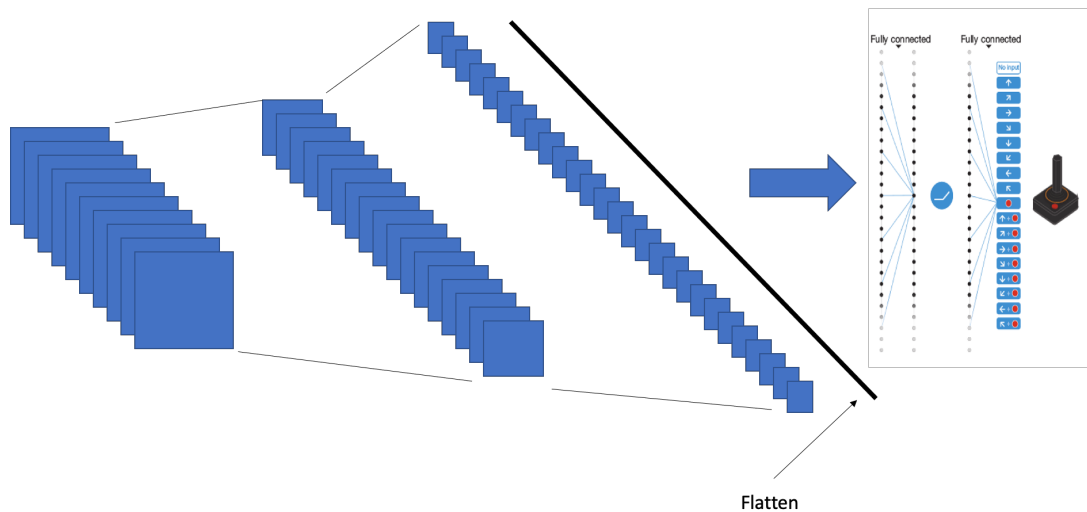


To speed up the training, you will create a DNN (as discussed in class) that will try to estimate the position of the ball and center position for the left and right players. For example, we can use the following network to achieve that.

Code submitted should enable me to run 10 matches and tell me the number of wins with possibility to turn on visualization.



Then you will freeze this network before the flattening and creating a network that will estimate the $\hat{Q}(s, a)$ with another dense network (this is called transfer learning).



After training the network, you will present a short movie training the network, and play 100 games against OpenAI player.

You will report.

Average number of steps until someone wins, and the score of the game.

Deliverables:

- Trained network that allows me to test what you did
- Average number of steps

- Short report 6 pages, explaining the architecture of the solution, hyperparameters you tried, etc

Your final grade for the exam will be a function of the number of steps until you win (or loose), and the score.