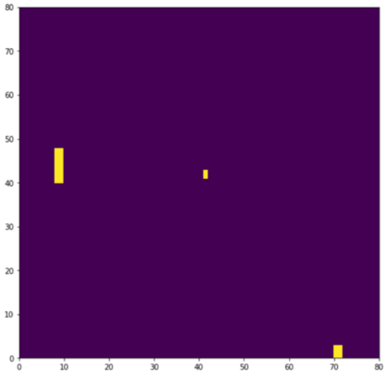
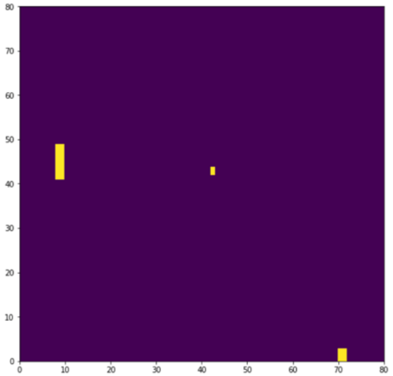
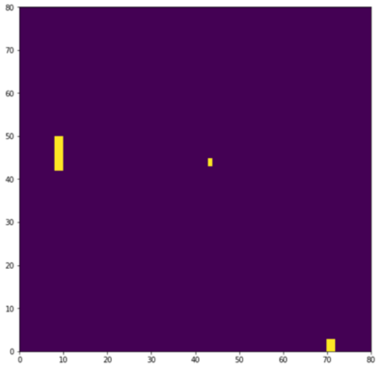
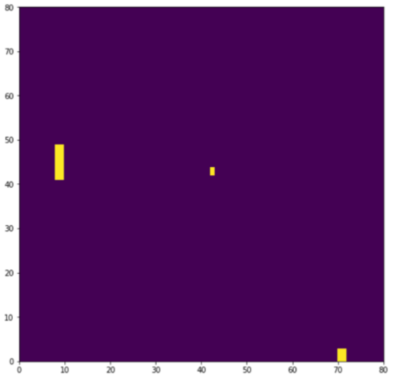
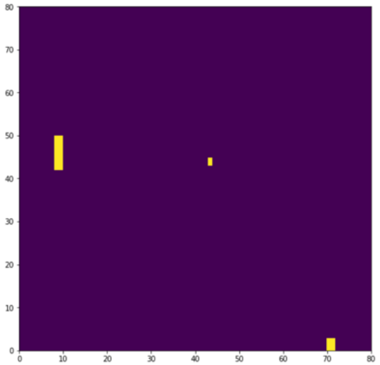
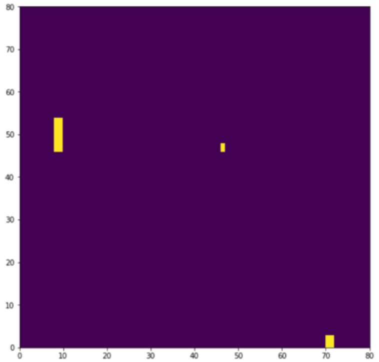
TOPIC 4 REINFORCEMENT LEARNING

TEXAS
The University of Texas at Austin

Deep Q-Learning

State	Frame 7	Frame 8	Frame 9	Button	Want to make small:
S_9				2	$(v_2(s_9) - r_9 - \delta \max\{v_0(s_{10}), v_2(s_{10}), v_3(s_{10})\})^2$

	Frame 8	Frame 9	Frame 10	Button	Want to make small
S_10				3	$(v_3(s_{10}) - r_{10} - \delta \max\{v_0(s_{11}), v_2(s_{11}), v_3(s_{11})\})^2$

Deep Q-Learning

- NN's work like regression
    - $\min \sum_t \left(predicted\ v(s_t) - true\ v(s_t)\right)^2$
- $predicted\ v(s_t)$ is like $\hat{y}$ in OLS
    - In training you just tell TF the set of s_t's
    - TF then tries to wiggle weights and biases to make predicted close to truth

Deep Q-Learning

- Technically, the code we saw earlier was a *Double* Deep Q-Network
- To be just a simple Deep Q-Network we would take an SGD step after each frame was played
- Double Deep Q-Networks use one network to estimate the truth, while learning on another network
  - Periodically update the truth giving network
  - This is exactly what we did: find the truth for every frame using the old network weights then run several SGD steps to update the weights

Improve Performance

- In PG we used the true discounted reward to evaluate our performance
- Could we do this in a variant of Q-learning
  - When generating the truth don't use $r_t + \max_x \delta v(S_{t+1}, t+1)$
  - Instead use the actual discounted reward at the end of the point, as in PG
- Who knows if this will be any better…give it a shot

Actor-Critic Methods

- One new strategy is to combine DQN with PG
- In PG we used the true discounted reward as our weight for the loss
- The actor-critic method uses the estimated value function from DQN as the weight for the loss function
  - Use PG to pick actions that get chosen
  - Use DQN to evaluate if the actions are good or not
    - Weight in the objective
- Train both networks simultaneously

Actor-Critic Methods

- This is advantageous because it helps both networks decouple acting and learning
- Both acting and learning can be more focused!
- Q-learning sometimes has a bias issue when you are training and using the NN to pick your action
- Policy gradients should look at the expected future reward, instead of the actual future reward of the particular sample path!

Dueling Networks

- A recent advance in RL is to train 2 NNs and have them play against each other
  - Dueling networks
- When NN1 makes a decision, it knows the distribution of NN2's actions at this state
- NN1 optimizes according to what it knows NN2 will do
- The same is true for NN2
- We'll see more of this when we go back to DP