CS 330 Autumn 2020 Homework 3

SUNet ID: 06009508

Name: Brandon McKinzie

Collaborators: N/A

By turning in this assignment, I agree by the Stanford honor code and declare that all of this is my own work.

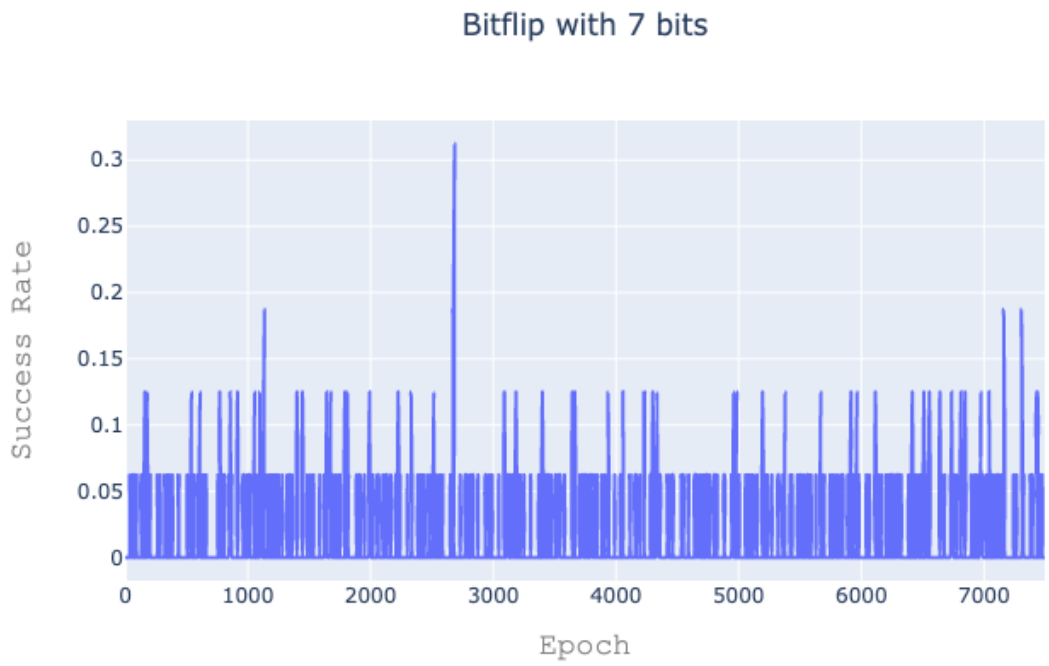
[LINK TO COLAB NOTEBOOK]

Raw URL: https://colab.research.google.com/drive/11yfl2CZRcITHXDl2FD94O3XZ57bkFzQj?usp=sharing

---

Problem 1: Implementing Goal-Conditioned RL on Bit Flipping

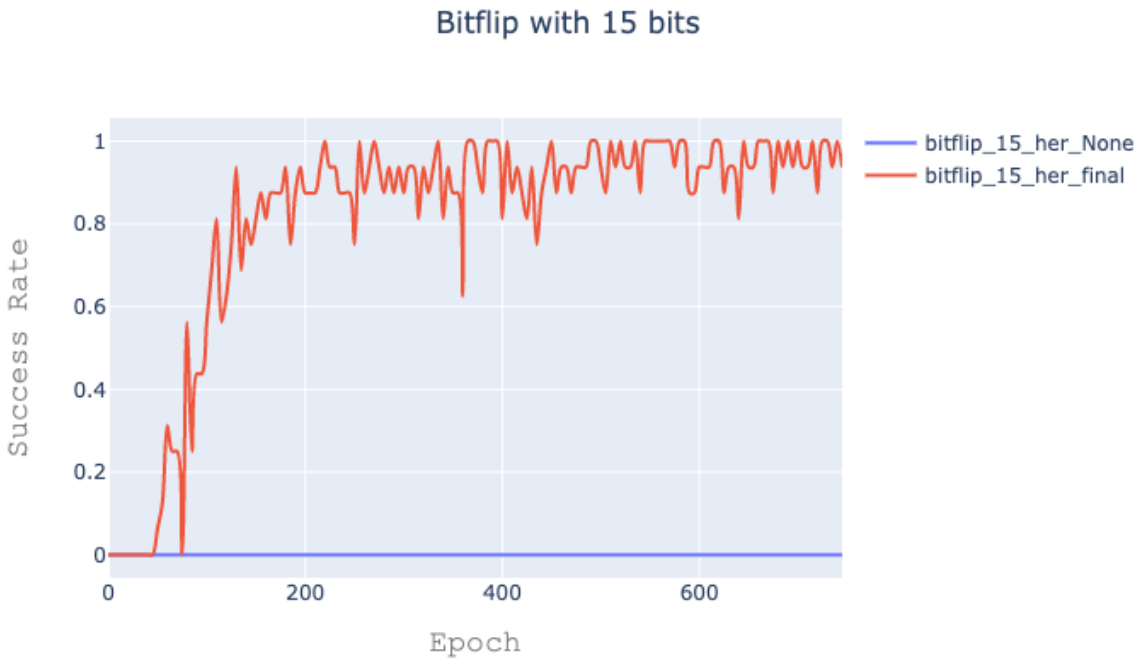
Part a. The figure below shows the success rate for 150 epochs without goal-conditioning or HER. Since the model has no way of conditioning its predictions on the desired goal state, it is essentially guessing randomly, which agrees with its success rate hovering on the order of $1/2^7$. No analysis was requested.



Problem 3: Bit Flipping Analysis

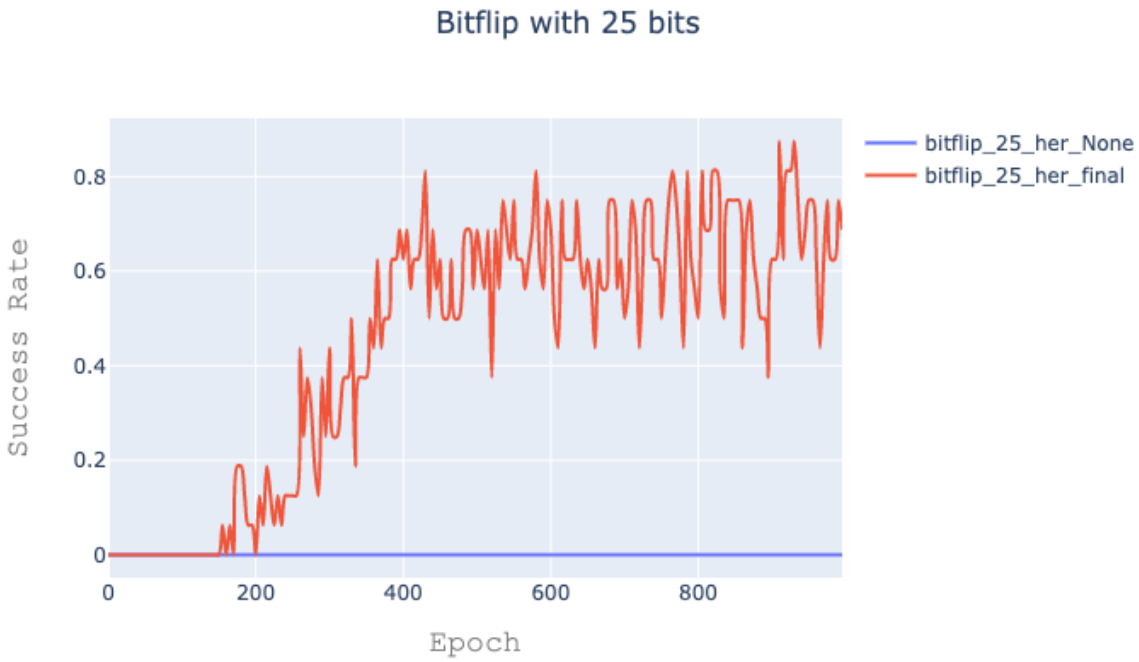
Part a. The results show how including hindsight experience replay enabled the model to reach perfect success rate in far fewer epochs than without HER. Without HER, the model essentially has to "wait" until it sees a reasonable number of positive examples (where it reached the goal) to have any hope of learning, which is inefficient in terms of progress per epoch.

Part b[1] Increasing the number of bits from 7 to 15 destroyed the performance of the model not using HER. This is because the probability of it randomly reaching the goal state is far lower than it was for the 7-bit case. This means it had nearly zero positive examples to learn from and thus made no progress. In contrast, the model with HER "final" still received a steady stream of positive examples each epoch, allowing it to reach a perfect success rate after roughly 200 epochs.



[1] I ran this for more epochs (750) than requested out of curiosity.

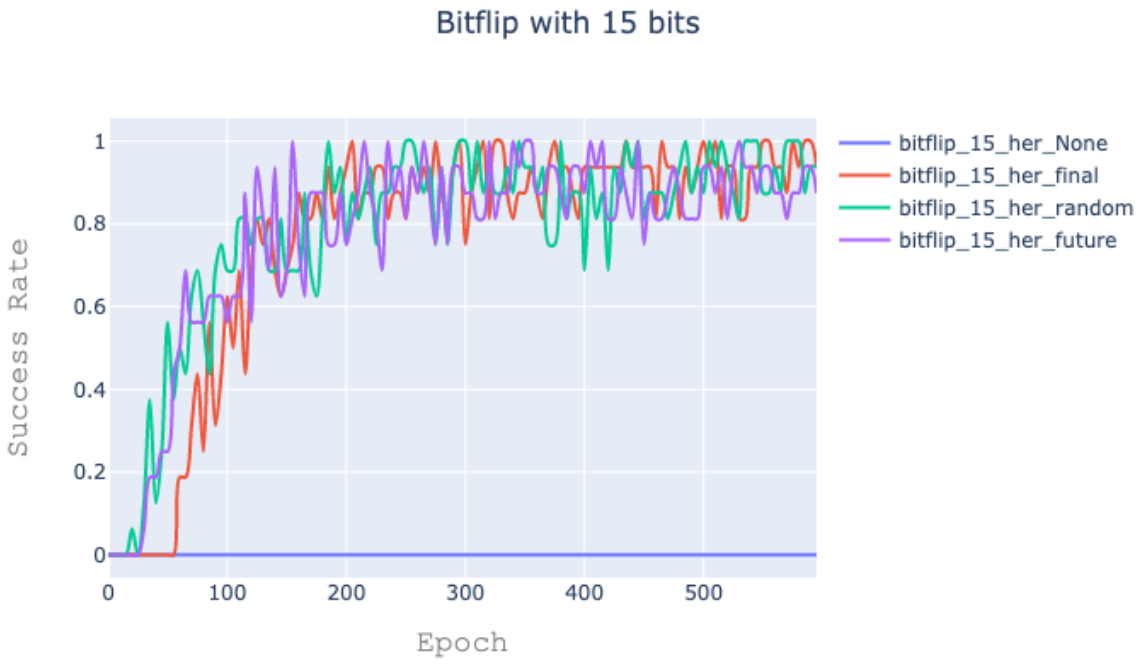
Part c. Unsurprisingly, increasing the number of bits to 25 still resulted in the HER "None" model making zero progress, since the probability of stumbling randomly onto a positive example decreases exponentially with the number of bits. Even for the HER "final" setting, most episodes are going to have the "final" state correspond to the 25th step in the episode[2], resulting in significantly more negative than positive examples (roughly 24:1).



NOTE: this plateaus around 0.7 due to a bug in the starter code. See piazza `594`. I did not have time to re-run my experiments with the fix (copying the state).

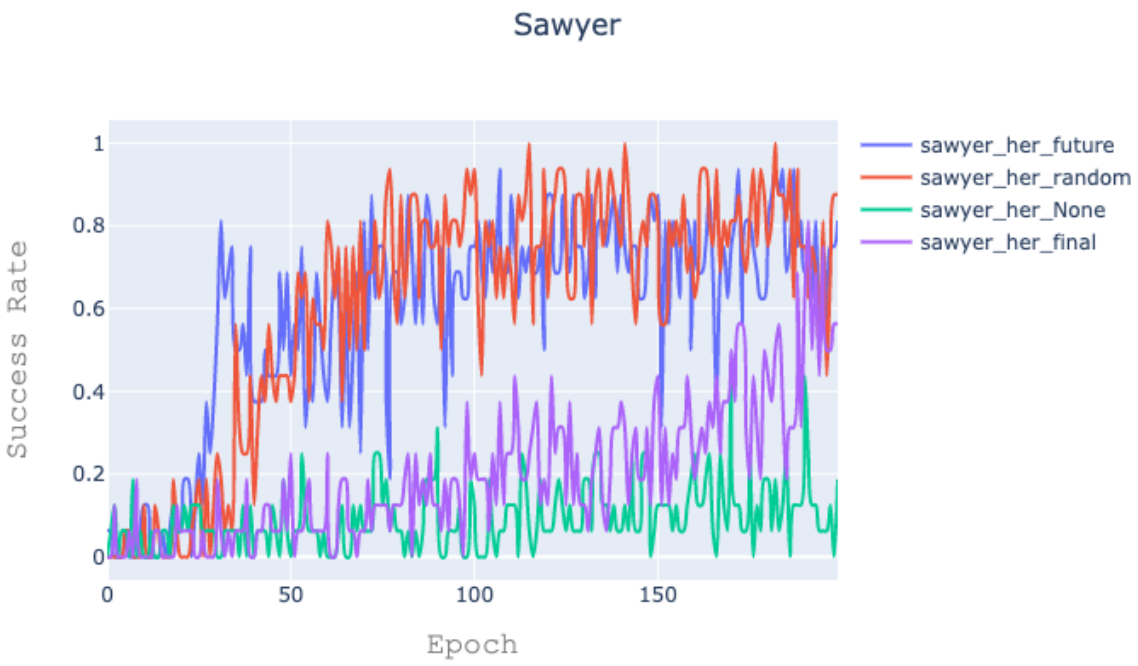
[2]Since reaching the true goal state randomly before the 25th step is $1/2^{25}$ which is virtually zero.

Part d. The random, future, and final settings for HER resulted in roughly similar performance, with random and future having better initial performance than final. Intuitively, the random and future settings are both more likely to encounter positive examples (as well as states that are nearby their associated goal states) compared to final, which makes learning easier particularly in the beginning.



Part e. See above for the 2-3 sentences for each part (in its associated part).

Problem 5: Sawyer Analysis



Similar to the bit flipping environment, HER set to "None" has the worst performance compared to the other settings. However, the "random" and "future" settings noticeably outperform the "final" setting here, whereas they were about the same for the bit flipping environment. This is primarily due to the differences in the reward function of the two environments.

In the Sawyer environment, the reward function is continuous and tells " how close" we are to the goal state, whereas the bit flipping reward function is just a binary yes/no on whether we've reached the goal state. For both "random" and "future", the model is provided more examples where it's nearby the goal state, compared to the "final" setting where it more often than not gets told "you're pretty far away." In a 2-dimensional environment, where distance from a goal is still ambiguous (an extra degree of freedom needs to be resolved such as angle), it is much easier to learn from examples where you're already relatively close to the goal state.

PS: I added a sanity check in my code to verify that the reward given to us by the Sawyer env was indeed the negative Euclidean distance between `s_` and `g`. However, this check was violated seemingly whenever $s\_\_ \approx g$, for whatever reason. The environment appears to change the reward slightly when we've landed at the goal state.