Lecture 19: Exploration

1 Introduction

Why exploration? Last time, we motivated exploration as being necessary to find new solutions to challenging problems, finding solutions that are better than the best solutions that humans have discovered. There's any reason to study exploration, going back to one of the motivations for RL we discussed at the beginning of the class: as a model of how humans and other animals make decisions. RL is convenient for describing how optimal decisions get made, and as both Prof. Niv and Prof. Pillow have mentioned, there's neural evidence that the brain is actually implementing some RL algorithm. However, there remains a pretty profound question here: what reward function is the brain maximizing with RL?

One natural choice is that it's maximizing fitness and survival, minimizing pain. While functionally useful, these motives fail to explain *play* that we observe. It seems like there is also some sort of intrinsic motivation or curiosity that drives animals to play and explore.

Plan for today

1. Bootstrap DQN – disentangling different sources of uncertainty.
2. Instrinic rewards
3. Skills
4. Howard Chen, 5th yr PhD student in NLP.

2 Review: Parameter Space Noise

Simply add noise to the policy network when collecting data: $\pi_{\theta+\epsilon}(a \mid s)$.

3 Bootstrap DQN [14]

Challenges:

- It's unclear how much noise to add for parameter space noise: noise in parameter space doesn't necessarily correspond to meaningful noise in function space.
- The degree of exploration doesn't necessarily decrease throughout the learning process.
- The "drive" for exploration is somewhat heuristic, and not directly tied to *learning*.

3.1 Bootstrap DQN

The key idea used in Bootstrap DQN is to learn an *ensemble* of learned Q networks: $Q_{\theta_1}(\cdot,\cdot), Q_{\theta_2}(\cdot,\cdot), \cdots$. That is, we learn k Q networks, each with the same architecture but different weights. The most important difference is that these weights are randomly initialized. In theory, you're supposed to update each member of the ensemble on a different subset of the data; in practice this doesn't matter much.

Intuitively, you can see that these Q networks should converge to all be the same after you've collected enough data and done enough gradient updates. This is a desirable property – it means that the degree of exploration decreases over time, eventually going to zero.

Formally, we're using this ensemble of Q values to disentangle *epistemic* vs *aleotoric* uncertainty. Epistemic uncertainty corresponds to "knowable unknowns," a type of uncertainty that can be reduced with more observations (e.g., what is in the closet). Aleotoric uncertainty corresponds to "unknowabe unknowns," a type of uncertainty that cannot be reduced further with more information (e.g., what side will this coin land on). Because epistemic uncertainty can be decreased, it is exactly this sort of uncertainty that we might want to use to drive exploration. Ensembles give us a convenient way of doing this, which is why many exploration methods end up making use of ensembles.

Extension: learning to adapt to any model in the ensemble [5].

Transition: But bootstrap DQN requires a reward function. How can we get entirely reward-free exploration?

4 Intrinsic Rewards

Another class of methods modify the RL reward function:

$$r(s,a) = r_{\text{task}}(s,a) + r_{\text{intrinsic}}(s,a). \tag{1}$$

There are many different choices for what to use as the intrinsic reward:

- Errors in a predictive model [16, 18].
- Surprise – minimizing the errors in a predictive model [3].
- Information gain – How much your predictive model is updated after observing a transition [9, 12].
- Errors in an inverse action model [17].
- Errors in predicting the output of a *random* neural network: $r_{\text{intrinsic}}(s) = \|f_{\theta_1}(s) - f_{\theta_2}(s)\|_2^2$. [4]
- (Negative) number of times that a state has been visited (or some approximation thereof) [2, 13, 15, 19].

5 Choosing an exploration method

Some considerations to think about when choosing an exploration method:

1. Does it do temporally-correlated exploration
2. Does the degree of exploration decrease after a transition has been seen many times before?
3. How does the method handle the *noisy TV problem*, where transitions are highly stochastic at one state?
4. How well does the method scale to high-dimensional settings?

6 Skill Learning

A final set of methods are those that learn not one exploratory policy but many policies. That is, we are optimizing a set of policies, $\{\pi(a \mid s,z) \mid z \sim p(z)\}$. These policies are typically optimized to be different from one another: each should visit a unique set of states. Algorithmically, this is done by learning a classifier that looks at a state and predicts which skill got you to that state: $p(z \mid s)$. The intrinsic reward is then the log probability of this classifier $r_{\text{intrinsic}}(s) = \log p(z \mid s)$.

Formally, you can show that these skill learning methods [1, 6, 8, 11] maximize the mutual information between the skill indicator z and the future states – you want the skill indicator to exert a high degree of influence over the states you visit in the future. These skill learning methods have been used in a wide range of applications [10], and have some nice theoretical guarantees [7].

7 Exploration within your data

While we've primarily been looking at how to collect more data, there's a sense in which RL algorithms also do exploration within the data they have already collected:

- reweighting. This is what REINFORCE does. It looks at all the trajectories, weights by their return, and then mimics. This means that higher-return strategies are "reinforced."
- stitching. This is what SARSA and Q-learning do. Then can piece together bits of previously-seen trajectories to create new solutions. This enables these algorithms to identify a solution that has never been demonstrated before. This is why we call these methods off-policy methods.

Both these forms of learning relate to the *data*: searching within the data or recombining it to find better solutions.

References

[1] Achiam, J., Edwards, H., Amodei, D., and Abbeel, P. (2018). Variational option discovery algorithms. *arXiv preprint arXiv:1807.10299*.

[2] Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. (2016). Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29.

[3] Berseth, G., Geng, D., Devin, C., Rhinehart, N., Finn, C., Jayaraman, D., and Levine, S. (2019). Smirl: Surprise minimizing reinforcement learning in unstable environments. *arXiv preprint arXiv:1912.05510*.

[4] Burda, Y., Edwards, H., Storkey, A., and Klimov, O. (2018). Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*.

[5] Clavera, I., Rothfuss, J., Schulman, J., Fujita, Y., Asfour, T., and Abbeel, P. (2018). Model-based reinforcement learning via meta-policy optimization. In *Conference on Robot Learning*, pages 617–629. PMLR.

[6] Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. (2018). Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*.

[7] Eysenbach, B., Salakhutdinov, R., and Levine, S. (2021). The information geometry of unsupervised reinforcement learning. *arXiv preprint arXiv:2110.02719*.

[8] Florensa, C., Duan, Y., and Abbeel, P. (2017). Stochastic neural networks for hierarchical reinforcement learning. *arXiv preprint arXiv:1704.03012*.

[9] Friston, K. (2009). The free-energy principle: a rough guide to the brain? *Trends in cognitive sciences*, 13(7):293–301.

[10] Ghafouri, N., Vardakas, J. S., Ramantas, K., and Verikoukis, C. (2024). Energy-efficient intra-domain network slicing for multi-layer orchestration in intelligent-driven distributed 6g networks: Learning generic assignment skills with unsupervised reinforcement learning. *arXiv preprint arXiv:2410.23161*.

[11] Gregor, K., Rezende, D. J., and Wierstra, D. (2016). Variational intrinsic control. *arXiv preprint arXiv:1611.07507*.

[12] Houthooft, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., and Abbeel, P. (2016). Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29.

[13] Lee, L., Eysenbach, B., Parisotto, E., Xing, E., Levine, S., and Salakhutdinov, R. (2019). Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*.

[14] Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. (2016). Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29.

[15] Ostrovski, G., Bellemare, M. G., Oord, A., and Munos, R. (2017). Count-based exploration with neural density models. In *International conference on machine learning*, pages 2721–2730. PMLR.

[16] Oudeyer, P.-Y. (2004). Intelligent adaptive curiosity: a source of self-development.

[17] Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR.

[18] Stadie, B. C., Levine, S., and Abbeel, P. (2015). Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*.

[19] Tang, H., Houthooft, R., Foote, D., Stooke, A., Xi Chen, O., Duan, Y., Schulman, J., DeTurck, F., and Abbeel, P. (2017). # exploration: A study of count-based exploration for deep reinforcement learning. *Advances in neural information processing systems*, 30.