Deep Reinforcement Learning

Xiyuan Yang
2025.11.22

Lecture Notes for Deep Reinforcement Learning, CS285

目录

§1. Introduction

§1.1. Works to Cover

1. From supervised learning to decision making
2. Model-free algorithms: Q-learning, policy gradients, actor-critic
3. Model-based algorithms: planning, sequence models, etc.
4. Exploration
5. Offline reinforcement learning
6. Inverse reinforcement learning
7. Advanced topics, research talks, and invited lectures

§1.2. Introduction to RL

§1.2.1. Supervised Learning

Given $D = \{(x_i, y_i)\}$, we want the supervised learning systems to learn how to predict y from x: $f(x) \approx y$.

It usually assumes:

- i.i.d data.(独立同分布)
- known ground truth outputs in training

For example, Deep Learning for Image Recognitions/Classifications. (**Need High-Labeled Data**)

§1.2.2. Reinforcement Learning

- Data is not i.i.d: previous outputs influence
- Ground truth answer is not known, only know

if we succeeded or failed, more generally, we know the reward

> **Recordings.**
> 强化学习对数据的利用更加的松弛，不需要高质量人工标注的数据，这提升了强化学习的上限，但是这也导致模型对数据的利用率较低。

In the mathematical view:

- goal for supervised learning: $f_{\theta(x_i)} = y_i$
  - training data $\{(x_i, y_i)\}$ are fixed and manually labeled.
- goal for reinforcement learning: learning $\pi_\theta : s_t \to a_t$ to maximize $\sum_t r_t$
  - the data $(s_1, a_1, r_1, ..., s_T, a_T, r_T)$: own actions, dynamic!

§1.2.2.1. Applications

- games, robotics
- RL with Large Language Models
- RL with image generations
- RL for chip design

§1.2.3. Deep Reinforcement Learning

Supervised Learning has the upper-bound has labeled data, but RL does not.

"Move 37" in Lee Sedol AlphaGo match: reinforcement learning "discovers" a move that surprises everyone.

- Data Driven AI (learns about the real world from data, but doesn't try to do better than the data)
- Reinforcement Learning (optimizes a goal with emergent behavior, but need to figure out how to use **at scale**).

Combination: **Deep Reinforcement Learning**!

> **Recordings** The Bitter Lesson.
> We have to learn the bitter lesson that building in how we think we think does not work in the long run.

- Data without optimization doesn't allow us to solve new problems in new ways.
- Optimization without data is hard to apply to the real world **outside of simulators**.

The core components (two general building blocks for AI-systems):

- Learning: use data to extract patterns (world laws), understanding the world
- Search: Use computations to extract inferences. Making inferences and leverages that understanding for emergence.

> **Recordings.**
> We have a brain for one reason and one reason only – that's to produce adaptable and complex movements. Movement is the only way we have affecting the world around us··· I believe that to understand movement is to understand the whole brain.

§1.2.4. Sequential Decision Making

Far more than a convex optimization problem!

- Learning reward functions from example (inverse reinforcement learning)
- Transferring knowledge between domains (transfer learning, meta-learning)
- Learning to predict and using prediction to act

real-time reward is hard to design.

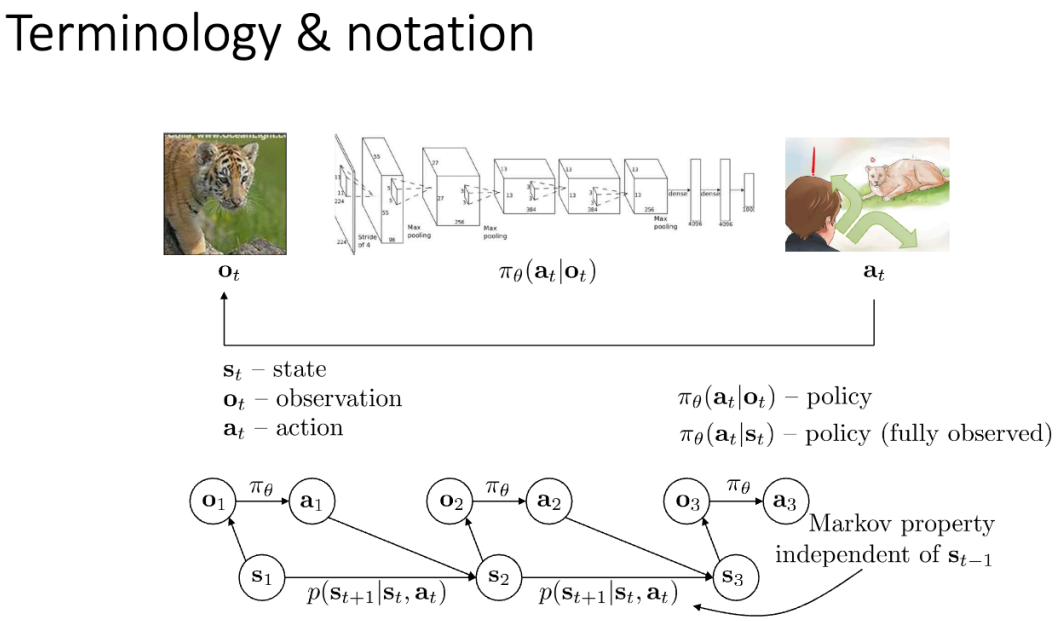
- Learning from demonstrations
  - Directly copying observed behavior
  - Inferring rewards from observed behavior (inverse reinforcement learning)
- Learning from observing the world
  - Learning to predict
  - Unsupervised learning
- Learning from other tasks
  - Transfer learning
  - Meta-learning: learning to learn

Will RL be the way to AGI? (using a **general learning algorithms** for interacting observations and actions with the environment)

§2. Supervised Learning of Behaviors

- policy based on observations: $\pi_\theta(a_t|o_t)$
- policy based on full observations: $\pi_\theta(a_t|s_t)$
- policy are distributions (probability)

We can form the bayes net.



Recordings Markov Properties.
If you get the state s_t, then that is all you need to compute future state, and $s_1, s_2, ..., s_{t-1}$ does not matter.

This gives the properties of the state.

§2.1. Imitation Learning

Target: Given the labeled data, trying to learn the $\pi_\theta(a_t|o_t)$ by supervised learning. Given the o_t and a_t, it forms the training data.

It is a kind of **behavior cloning**.

这样的问题在于模型只会模仿先验的正确答案，而一旦预测出现微小的偏差，这一部分的偏差就会不断的方法，导致在多个时间步后模型的状态发生较大的偏移。

例如，自动驾驶的三个前置摄像头可以保证一定的鲁棒性的提升。

Core reason: **i.i.d** assumption does not work!

- Data augmentation for Training Data
- Algorithms Change
- Multi-Task Learning

§2.1.1. Theory

For the training loops, the distributions for the training data is $p_{\text{data}}(o_t)$, which is different from the distributions of the testing environment p_{π_θ}. Thus when trained model encountered new observations that not appears in the training set, it will cause the bias.

So how can we define a good policy? Assume **GT behavior** for expert is deterministic if given the whole observations s_t. (Just for simplify), and we can define the cost functions:

$$c(s_t, a_t) = \begin{cases} 0 & \text{if } a_t = \pi^*(s_t) \\ 1 & \text{otherwise} \end{cases}$$

And our goal is to minimize:

$$\mathbb{E}_{s_t \sim p_{\pi_\theta}(s_t)}[c(s_t, a_t)]$$

注意！这里在训练是的分布就是 π_θ 相当于模型直接在训练过程中进行分布的采样，目标是最小化策略在自身轨迹上执行动作与专家动作不一致的概率。

- This is some kind of **Dataset Aggregation**.
- **Attention**! It is the state not the observations! We need to use the Markov properties for state.

Assume supervised learning works:

$$\pi_\theta(a \neq \pi^*(s)|s) \leq \varepsilon, \forall s \in \mathcal{D}_{\text{train}}$$

在训练数据分布下，模型动作和专家动作不一致的概率不超过 ε.

$$E\left[\sum_t c(s_t, a_t)\right] \leq \varepsilon T + (1-\varepsilon)(\varepsilon(T-1) + (1-\varepsilon))(...)) = O(\varepsilon T^2)$$

This will leads to the cascading errors with many many time steps. (For the worse case.)

More generally:

$$p_\theta(s_t) = (1-\varepsilon)^t p_{\text{train}}(s_t) + (1-(1-\varepsilon)^t) p_{\text{mistake}}(s_t)$$

For the main distributions p_{mistake}, we don't see them in the training data.

$$|p_\theta(s_t) - p_{\text{train}}(s_t)| = (1-(1-\varepsilon)^t)\ |p_{\text{mistake}}(s_t) - p_{\text{train}}(s_t)| \leq 2(1-(1-\varepsilon)^t) \leq 2\varepsilon t$$

Thus:

$$\begin{aligned} \sum_t P_{p_\theta(s_t)}[c_t] = \sum_t \sum_{s_t} p_\theta(s_t) c_t(s_t) &\leq \sum_t \sum_{s_t} p_{\text{train}}(s_t) c_t(s_t) + |p_\theta(s_t) - p_{\text{train}}(s_t)|\ c_{\max} \\ &\leq \sum_t \varepsilon + 2\varepsilon t \end{aligned}$$

Recordings.

- In reality, we can recover from mistakes.
- A paradox: imitation learning can work better if the data has more mistakes (and recoveries)!
- The imitation learning:
  - Teach the models when the models are on the right way.
  - **Teach the models to recover when the models are outside the right way.**

§2.1.2. Data Augmentation

- Intentionally add mistakes and corrections (The mistakes hurt, but the corrections help, often more than the mistakes hurt)
- Use **data augmentation**. (e.g. side-facing cameras.), add some "fake" data that illustrates corrections.

§2.1.3. More Powerful Models

Recordings.
- Non-Markovian behavior ($\pi_\theta(a_t|o_1, o_2, ..., o_t)$)
  - Using histories
- Multimodal behavior

§2.1.3.1. Using Histories

Using the sequence model (LSTM, transformers.)

However, learning from history may cause confusions: (ICLR-2019 Best Paper)

Behavior Cloning will only learn about the correlations, but not the **cause and effect**, which is mortal in autonomous driving. (**casual confusion**)

Solutions: data augmentation & diffusion models.

§2.1.3.2. Multimodal behavior

- mixture of Gaussians

$$\pi(a|o) = \sum_i w_i \mathcal{N}\left(\mu_i, \sum_i\right)$$

More specifically, it can be written into:

$$\pi(a|o) = \sum_{k=1}^{K} \pi_k(o) N\left(a|\mu_k(o), \sum_k(o)\right)$$

$\mu_k(o)$ and $\sum_k(o)$ means they are functions of given input observations, but not sharing the same global parameters.

- latent variable models (conditional variant auto-encoder)

Recordings.
Latent variable models (潜变量模型) 在模仿学习中的核心原理是: 用一个低维的、简单分布的潜变量 z, 把专家的多模态连续动作分布 $p(a|s)$变得"可解析、可高效采样、可无限表达"。

$$\pi(a|s) = \int p(a|z,s)p(z|s)\mathrm{d}z \approx \int \mathcal{N}(a|f_\theta(z,s), \delta^2\mathbb{I})\mathcal{N}\left(z|g_\varphi(s,a), \sum\right)\mathrm{d}z$$

- diffusion models

Diffusion models for image generations: $f(x_i) = x_i - x_{i-1}$

For imitation learning:

- $a_{t,0}$ is the true actions
- $a_{t,i+1} = a_{t,i} + \text{noise}$
- $a_{t,i-1} = a_{t,i} - f(s_t, a_{t,i})$
- Autoregressive discretization (like the sequence language models)

Do predictions: $p\left(a_{t,i}|s_t, a_{t,1}, a_{t,2}, ..., a_{t,(i-1)}\right)$ (discretize one dimension at a time)

To multiply together: $p(a_t|s_t)$, like the complexity in the model distributions.

Recordings.
- 使用自回归分解来逐步离散化高维空间的正确性在于，只要维度之间存在任何相关性，自回归分解都能表达它。
- 并且自回归分解就像 token 生成一样，是一个信息不断增益的过程，不会对原有的信息造成影响。并且这样做离散化可以极大的减少状态空间的个数 $(K^D \to KD)$
- 类似于语言模型逐 token 的生成，不限制句子的长度实现了最终的 adaptive compute 并且每一次输出的维度限制在了词表的维度。

§2.1.4. Multi-Tasking Learning

§2.1.5. DAgger

§3. Conclusion