Reinforcement Learning

Lecture 10: Extended methods

Chris G. Willcocks Durham University

Lecture overview



1

Distributed and recurrent RL

- DQN characteristics
- recurrent replay in distributed RL
- R2D2 performance

2 Exploration vs exploitation

• approaches

3 Intrinsic rewards

• NGU: intrinsic motivation and curiosity

4 Latent recurrent imagination

• Dreamer and DreamerV2

5 AlphaStar and looking forward

- starting supervised
- self-play and league-play



Characteristics: DQN

DQNs optimise a function (neural network) to predict the *Q*-value (the expected reward) for a given state and aciton.

$$\hat{q}_{\mathbf{w}} \rightarrow \hat{q}(s_t, a_i, \mathbf{w})$$

DQN doesn't work very well for long-term credit assignments:



Example: function approximation

The function can be approximated:

Q = np.zeros([n_states, n_actions])
a_p = Q[s,:]

action—value table is approximated: a_p = DeepNeuralNetwork(s)

This usually requires a few extra tricks to work:

- Double DQN
- Prioritised replay
- Distributed RL



Definition: R2D2

Recurrent Replay Distributed DQN (R2D2) [1] uses RNNs, training on a sequence of m = 80 observations o_t and hidden states h_t :

Computation of riangle Q



Therefore it can backpropagate through the history, updating where earlier actions led to long-term future reward.

Definition: distributed RL

In distributed RL [2, 3], a central learner (with some parameters θ) receives experience from multiple parallel workers $w_1, w_2, ..., w_n$ which run episodes independently:





These graphs shows R2D2 performance for $\gamma = 0.99$ (pink) vs $\gamma = 0.997$ (orange):



Watch R2D2 play Gravitar 🗹

Watch R2D2 play other Atari 🗹

Exploration vs exploitation approaches



Exploration vs exploitation

R2D2 is not good at balancing exploration vs exploitation. There are other exploration strategies besides taking random actions:

- random exploration, as before:
 - ε-greedy
 - softmax
- optimisim in the face of uncertainty
 - estimate uncertainty of the value
 - prefer exploring states/actions with higher uncertainty
- information state space
 - the agent information is part of the state description
 - quantifies state information value

Exploration in Gravitar and AoE

Randomly choosing isn't always good:









Intrinsic rewards NGU: intrinsic motivation and curiosity



Definition: intrinsic reward

Never Give Up (NGU) [4] extends R2D2 by adding an intrinsic reward R', which is where the agent adds its own reward on top of the environment reward:

 $R_t = R_t^{\mathsf{env}} + \beta R_t',$

where β weights the exploration according to its intrinsic reward (e.g. curiosity).

Specifically, it adds a reward for finding things that it has not yet seen before.

- intrinsic motivation
- curiosity
- novelty

Example: intrinsic reward





Dreamer [5] and DreamerV2 [6] use a recurrent neural network to 'imagine' and plan ahead, all in the latent (feature representation) space:



(a) Learn dynamics from experience

(b) Learn behavior in imagination

(c) Act in the environment

Latent recurrent imagination Dreamer algorithm



Initialize dataset D with S random seed episodes. Initialize neural network parameters θ, ϕ, ψ randomly. while not converged **do**

for update step c = 1..C do

// Dynamics learning Draw *B* data sequences $\{(a_t, o_t, r_t)\}_{t=k}^{k+L} \sim \mathcal{D}$. Compute model states $s_t \sim p_{\theta}(s_t \mid s_{t-1}, a_{t-1}, o_t)$. Update θ using representation learning. // Behavior learning

Imagine trajectories $\{(s_{\tau}, a_{\tau})\}_{\tau=t}^{t+H}$ from each s_t . Predict rewards $\mathbb{E}(q_{\theta}(r_{\tau} | s_{\tau}))$ and values $v_{\psi}(s_{\tau})$. Compute value estimates $V_{\lambda}(s_{\tau})$ via Equation 6. Update $\phi \leftarrow \phi + \alpha \nabla_{\phi} \sum_{\tau=t}^{t+H} V_{\lambda}(s_{\tau})$. Update $\psi \leftarrow \psi - \alpha \nabla_{\psi} \sum_{\tau=t}^{t+H} \frac{1}{2} ||v_{\psi}(s_{\tau}) - V_{\lambda}(s_{\tau})||^2$.

 $\begin{array}{l} // \text{ Environment interaction} \\ o_1 \leftarrow \text{env.reset ()} \\ \text{for time step } t = 1..T \text{ do} \\ \\ \text{ Compute } s_t \sim p_\theta(s_t \mid s_{t-1}, a_{t-1}, o_t) \text{ from history.} \\ \text{ Compute } a_t \sim q_\phi(a_t \mid s_t) \text{ with the action model.} \\ \text{ Add exploration noise to action.} \\ r_t, o_{t+1} \leftarrow \text{env.step } (a_t) . \end{array}$

Add experience to dataset $\mathcal{D} \leftarrow \mathcal{D} \cup \{(o_t, a_t, r_t)_{t=1}^T\}.$

Model components

Representation	$p_{\theta}(s_t \mid s_{t-1}, a_{t-1}, o_t)$
Transition	$q_{ heta}(s_t \mid s_{t-1}, a_{t-1})$
Reward	$q_{ heta}(r_t \mid s_t)$
Action	$q_{\phi}(a_t \mid s_t)$
Value	$v_{\psi}(s_t)$

Hyper parameters

Seed episodes	S
Collect interval	C
Batch size	B
Sequence length	L
Imagination horizon	H
Learning rate	α

AlphaStar and looking forward starting supervised

Exploration vs exploitation

AlphaStar [7] uses many components, supervised learning, and league-play.











Summary

In summary:

- learn the foundations and concepts of the field, so you can speak the lingo...
- ...but you may want to approach overly complex papers more like an engineer
 - run the code and dismantle it back down to the concepts that make it work
- sample efficiency is an issue, which can be traded for with model-based imagination
- RL can have very high variance, try to lower it so you can iterate faster without training

References I



- [1] Steven Kapturowski et al. "Recurrent experience replay in distributed reinforcement learning". In: International conference on learning representations. 2018.
- [2] Lasse Espeholt et al. "SEED RL: Scalable and Efficient Deep-RL with Accelerated Central Inference". In: International Conference on Learning Representations. 2020.
- [3] Dan Horgan et al. "Distributed Prioritized Experience Replay". In: International Conference on Learning Representations. 2018.
- [4] Adrià Puigdomènech Badia et al. "Never Give Up: Learning Directed Exploration Strategies". In: International Conference on Learning Representations. 2020.
- [5] Danijar Hafner et al. "Dream to Control: Learning Behaviors by Latent Imagination". In: International Conference on Learning Representations. 2020.
- [6] Anonymous. "Mastering Atari with Discrete World Models". In: Submitted to International Conference on Learning Representations. 2021.
- [7] Oriol Vinyals et al. "Grandmaster level in StarCraft II using multi-agent reinforcement learning". In: <u>Nature</u> 575.7782 (2019), pp. 350–354.