Module 5

Deep Reinforcement Learning for NLP

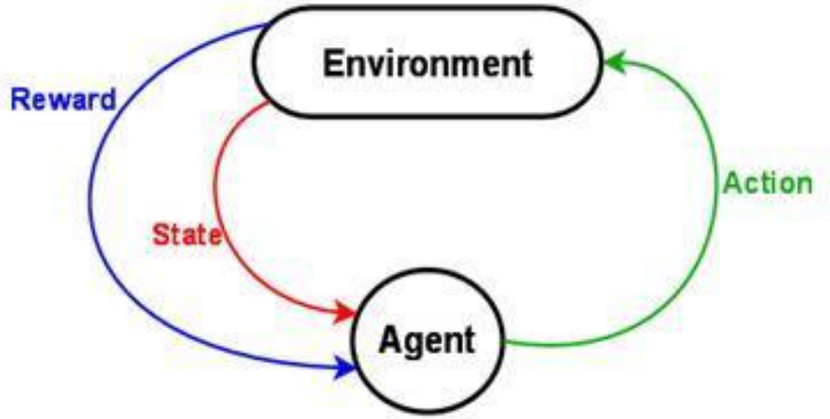
Deep reinforcement learning for NLP

- DRL Background
- DRL for NLP

Background of reinforcement learning

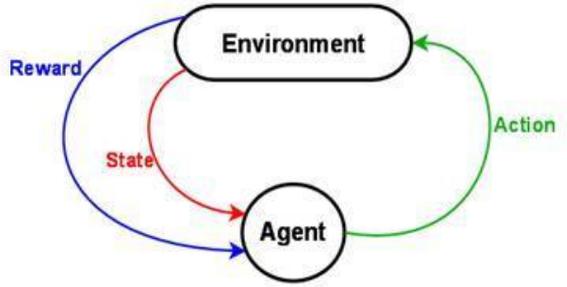
Reinforcement learning model:

- environment state set: *S*
- Action set: *A*
- rules of transitioning between states
- rules that determine the immediate reward of a state transition
- rules that describe what the agent observes



Sutton, Richard S.; Barto, Andrew G. (1998). Reinforcement Learning: An Introduction. MIT Press.

Q-Learning



Used to learn the policy of RL

Policy: a rule that the agent should follow to select actions given the current state

Q-Learning: find optimal policy for Markov decision process (MDP).

Approach: learning an action-value function, a.k.a. Q-function, that computes the expected utility of taking an action in a state – after training converges.

$$Q^{\pi}(s,a) = \mathbb{E}\left\{\sum_{k=0}^{+\infty} \gamma^k r_{t+k} \middle| s_t = s, a_t = a\right\},$$

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \eta_t \cdot \left(r_t + \gamma \cdot \max_a Q(s_{t+1},a) - Q(s_t,a_t)\right)$$

Watkins and Dayan, (1992), 'Q-learning.' Machine Learning.

Recent success

- Deep Q-Network (DQN)

1. Task: playing Atari games
2. RL setting: **huge state space**, e.g., raw image pixels from screen shots. But **small action space**, e.g., possible move of the joystick.
3. Model: using convolutional neural networks to compute $Q(s, a)$.
4. Results: achieve human level performance

action

state

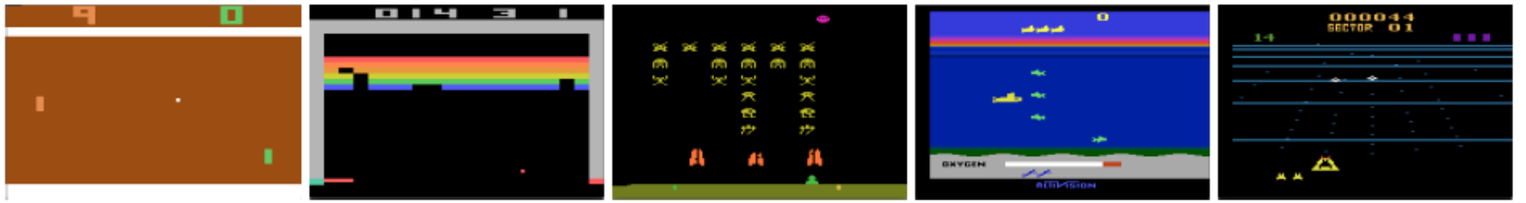


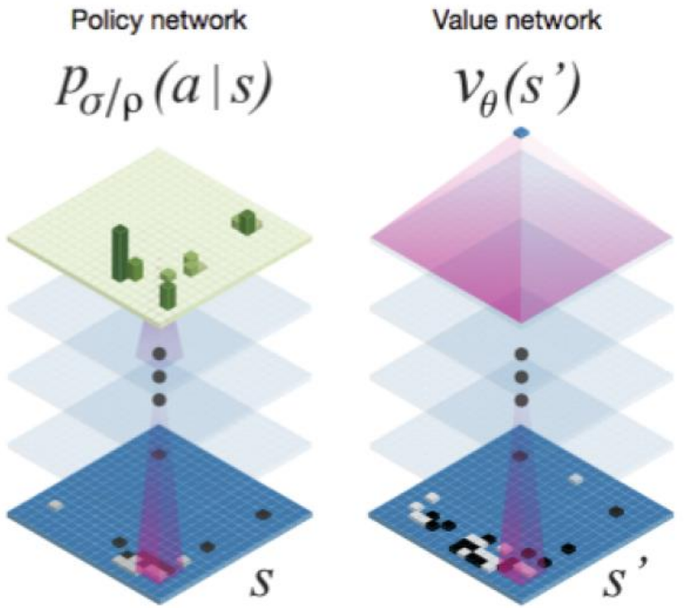
Figure 1: Screen shots from five Atari 2600 Games: (*Left-to-right*) Pong, Breakout, Space Invaders, Seaquest, Beam Rider

Mnih, Kavukcuoglu, Silver, Graves, Antonoglou, Wierstra, Riedmiller, "Playing Atari with Deep Reinforcement Learning", 2013

Recent success (cont.)

- AlphaGo

1. Task: playing Go
2. RL setting: **huge state space**, e.g., 19x19 board (highly complex and sensitive). But still **relatively small action space**, e.g., possible move (one out of <361 positions).
3. Model: built two deep networks: policy network and value network, both are CNNs
4. Use MCTS to estimate the value of states in a search tree.
5. Results: beat world Go Champion



Silver et al, "Mastering the game of Go with deep neural networks and tree search", 2016

Reinforcement learning for language understanding

- Consider the sequential decision making problem for text understanding:
  - E.g., Conversation, Task completion, Playing text-based games…
  - At time t:
    - Agent observes the state as a string of text , e.g., state-text s_t
    - Agent also knows a set of possible actions, each is described as a string text, e.g., action-texts
    - Agent tries to understand the "state text" and all possible "action texts", and takes the **right** action – right means maximizing the long term reward
    - Then, the environment state transits to a new state, agent receives a immediate reward.

[Narasimhan, Kulkarni, Barzilay. 2015]
[He, Chen, He, Gao, Li, Deng, Ostendorf, 2015]

Unbounded action space in RL for NLP

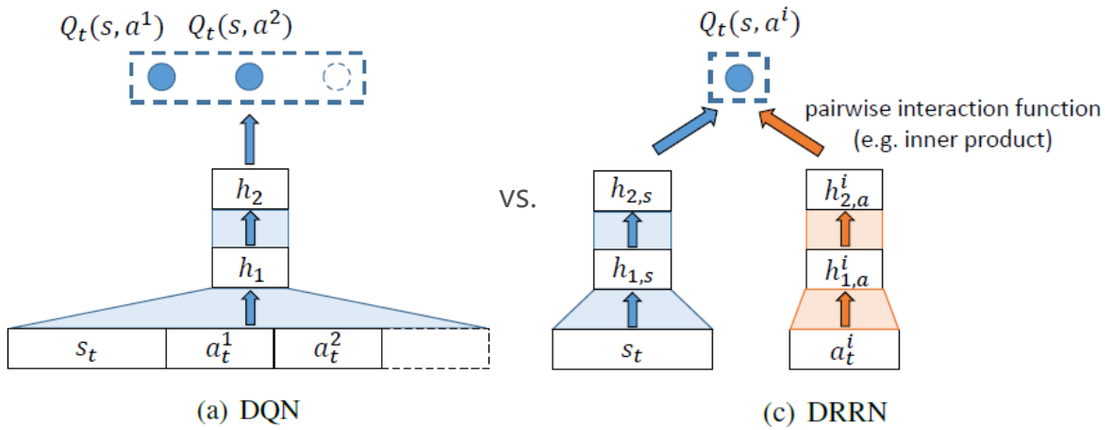
Not only the state space is huge, the action space is huge, too.

– Action is characterized by unbounded natural language description.

Example: a snapshot of a text-based game

Deep Reinforcement Relevance Networks

- Prior DQN work (e.g., Google DeepMind's work on Atari game, AlphaGo): state space unbounded, action space bounded.
- In NLP tasks, usually **the action space**, characterized by natural language, is discrete and nearly **unbounded**.
- We proposed a Deep Reinforcement Relevance Network (**DRRN**)
  - Project both the state and the action into a continuous space
  - Q-function is an relevance function of the state vector and the action vector



Eval metric	Average reward		
hidden dimension	20	50	100
NN-RL (2-hidden)	0.2 (1.2)	2.6 (1.0)	3.6 (0.3)
DQN (2-hidden)	2.5 (1.3)	4.0 (0.9)	5.1 (1.1)
DRRN (2-hidden)	7.3 (0.7)	8.3 (0.7)	**10.5 (0.9)**

Results on text-based game (reward higher the better)

[He, Chen, He, Gao, Li, Deng, Ostendorf, "Deep Reinforcement Learning with a Natural Language Action Space," 2015]

Visualization of the learned continuous space

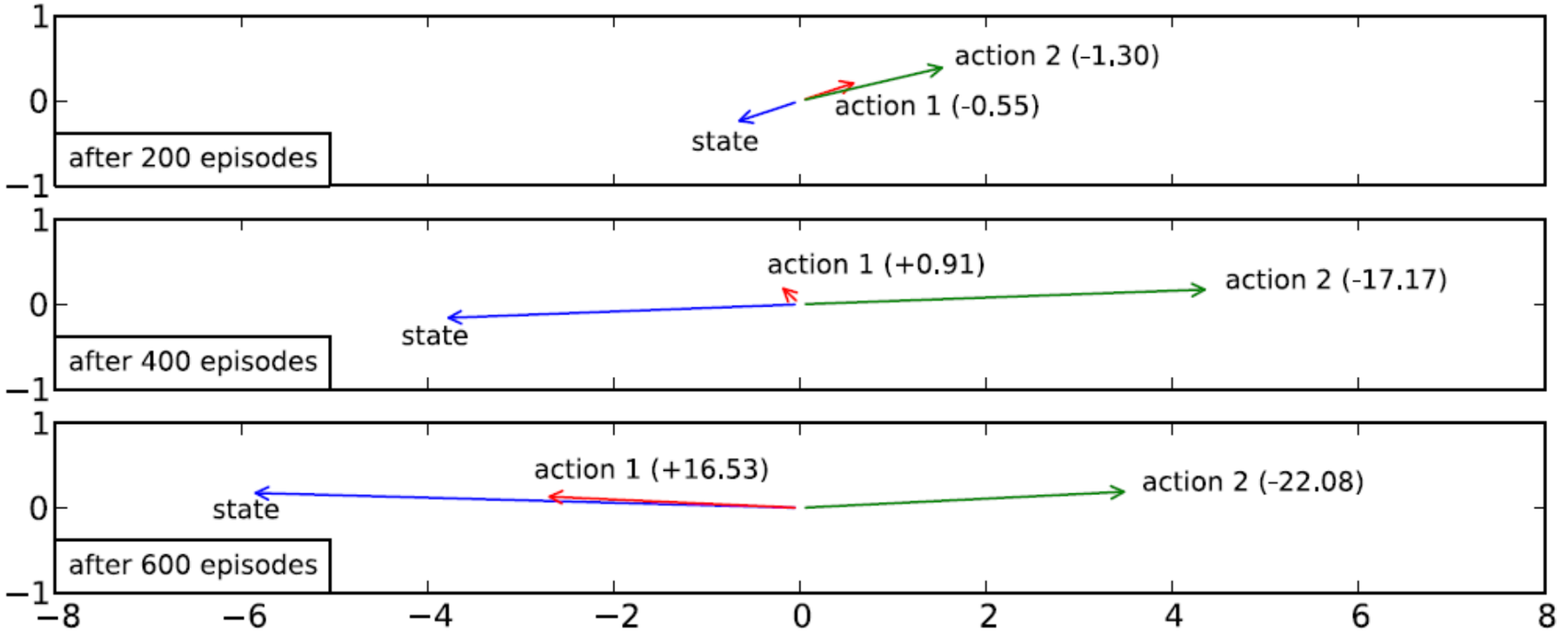
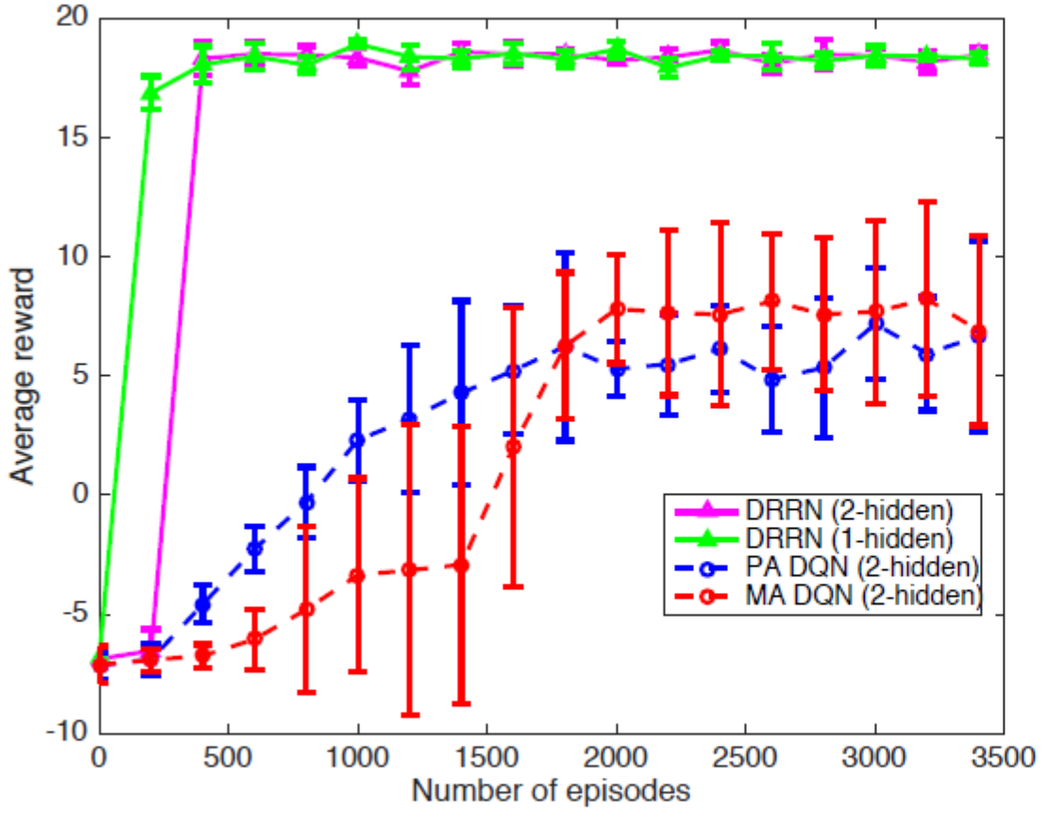
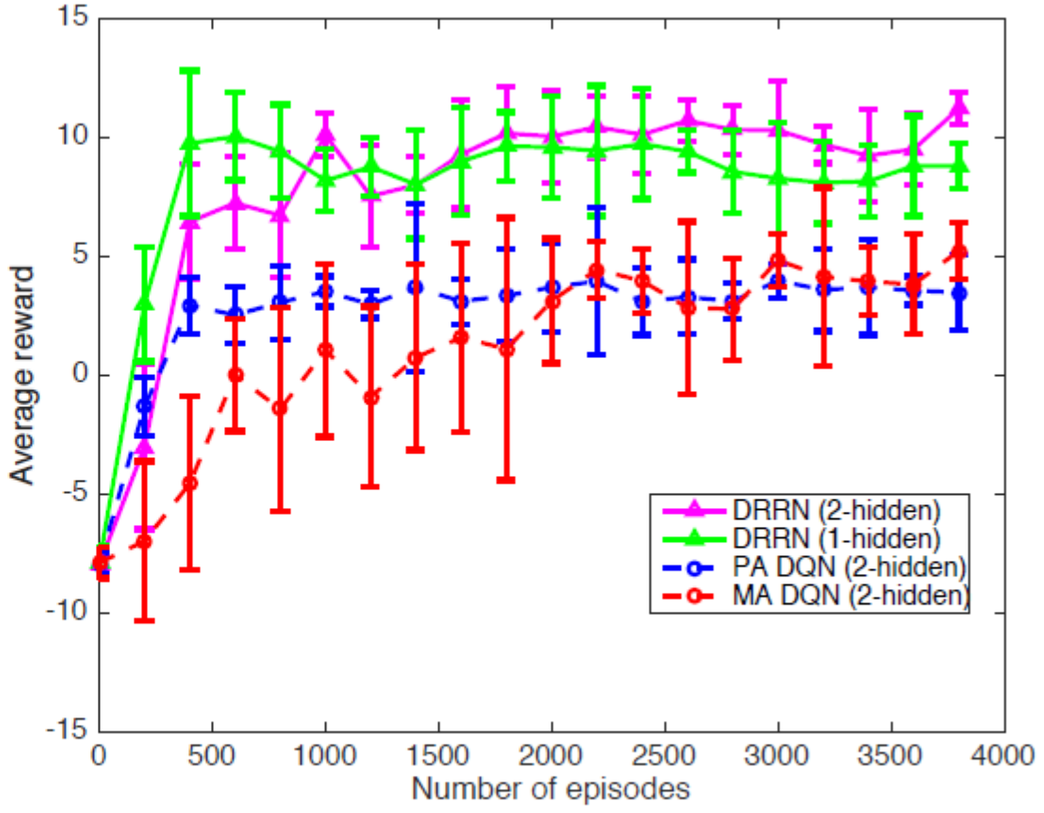


Figure 2: PCA projections of text embedding vectors for state and associated action vectors after 200, 400 and 600 training episodes. The state is "As you move forward, the people surrounding you suddenly look up with terror in their faces, and flee the street." Action 1 (good choice) is "Look up", and action 2 (poor choice) is "Ignore the alarm of others and continue moving forward."

Learning curve: DRRN vs. DQN



(a) Game 1: "Saving John"



(b) Game 2: "Machine of Death"

Tested on two text games

Q-function example values after converged

	Text (with predicted Q-values)
State	As you move forward, the people surrounding you suddenly look up with terror in their faces, and flee the street.
Actions in the original game	Ignore the alarm of others and continue moving forward. (-21.5) Look up. (16.6)
Paraphrased actions (not original)	Disregard the caution of others and keep pushing ahead. (-11.9) Turn up and look. (17.5)
Positive actions (not original)	Stay there. (2.8) Stay calmly. (2.0)
Negative actions (not original)	Screw it. I'm going carefully. (-17.4) Yell at everyone. (-13.5)
Irrelevant actions (not original)	Insert a coin. (-1.4) Throw a coin to the ground. (-3.6)

Note that, the DRRN generalizes to unseen actions well, e.g., for these "not original" actions, the model still gives a proper estimate of the Q-value.

Reinforcement learning for NLP tasks in a continuous space

- Project both states and actions (defined by *unbounded* NL) to a continuous semantic space using deep neural nets
- Compute the Q function in the continuous semantic space

- Open problems
  - Design of reward functions for specific tasks (e.g., conversation)
  - Handle combinatorial action spaces
  - Simulator modeling in NL tasks