Model Free Episodic Control

Nan Rosemary Ke

Problem formulation

- Current Reinforcement Learning algorithm learns very slow :
  - many iterations are needed
  - Slow learning due to gradient methods used
- Human can learn from only a few interactions
  - Very fast learning supported by hippocampus and medial temporal structure
- How to learn faster in machines?

How humans perform fast learning

Multiple systems for learning and memories

- Best learning scenario
    - Accurate model of the world
    - model -based planning
- Fast scenario
    - Learning new environment in model-free system
    - Model-free episodic control

Episodic Controller

Tabular approach. Given a state s, replay action that gave highest return.

$$Q^{EC}(s_t, a_t) = \begin{cases} R_t, & if(s_t, a_t) \notin Q^{EC}(s_t, a_t) \\ \max(Q^{EC(s_t, a_t)}, R_t) & otherwise \end{cases}$$

Issues with this approach

- Large memory needed for large problems
- Lack generalization to similar states

Episodic Controller

To solve the 2 issues.

- Size of memory: deleted least recently updated entry
- Generalization:
  - Novel states: non-parametric nearest neighbor model. For novel state s :

$$\widehat{Q^{EC}}(s,a) = \begin{cases} \frac{1}{k}\sum_{i=1}^{k} Q^{EC}(s_i,a), & if(s,a) \notin Q^{EC} \\ Q^{EC}(s,a), & otherwise \end{cases}$$

Episodic Controller

Algorithm

- For each episode:
  1. For t = 1, 2, ...T do:
     - Receive observation o_t from environment
     - Let s_t = gamma(o_t)
     - Estimate return for each action via (1).
     - Let a_t = argmax_a Q^EC(s_t, a)
     - Take action a_t, receive reward r_{t+1}
  - End For
  - For t = T, T-1, ..., 2, 1 do:
     - Update Q^{EC}(s_t, a_t) according to (1)
  - End For
- End For

Representation and Biological Plausibility

Biological plausibility:

- Hippocampus in the brain operates on representation which includes the output of the ventral systems, which is suppose to generalize

Implementation details:

- Original observation space is not practical, requires too much memory.
- We consider 2 different embedding
    - **Project** of original space into smaller dimensional spaces. Useful when only small changes occur.
    - Use **VAEs (variational autoencoder)** to map high dimensional data small dimensional data. Useful when many dimensions in the original spaces are useless.

Experiments

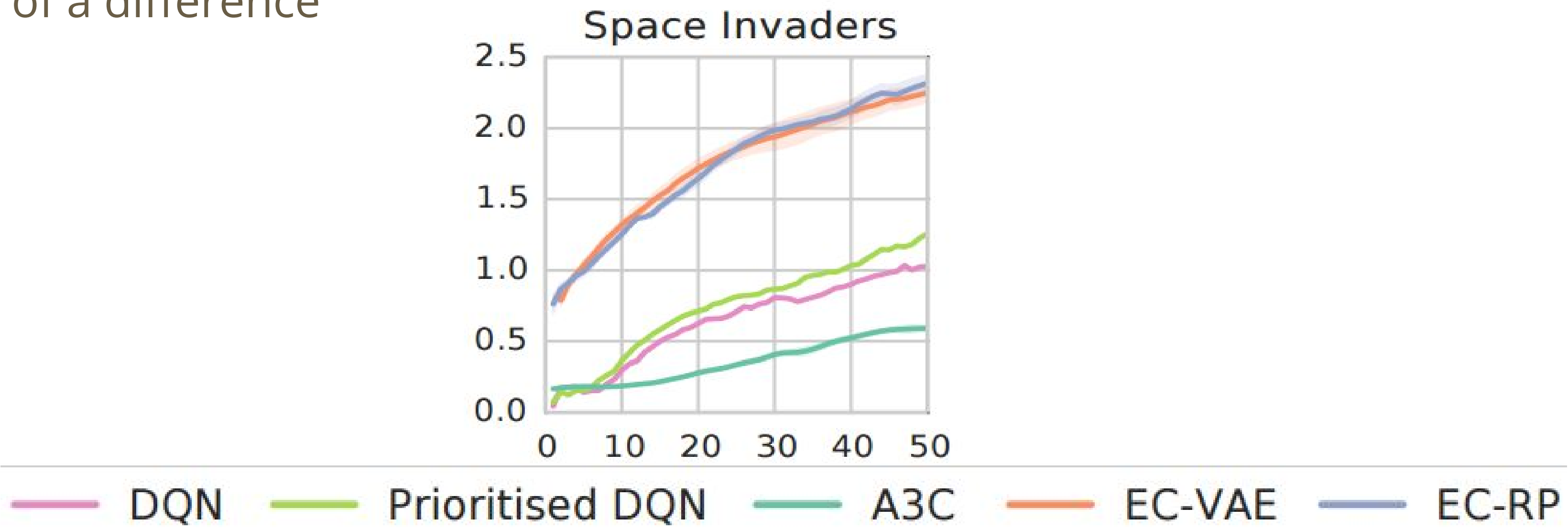
Atari game details

- Size of buffer: 1,000,000 entries.
- K nearest neighbor: k = 11
- Discount rate 1
- Epsilon greedy 0.005

Experiments

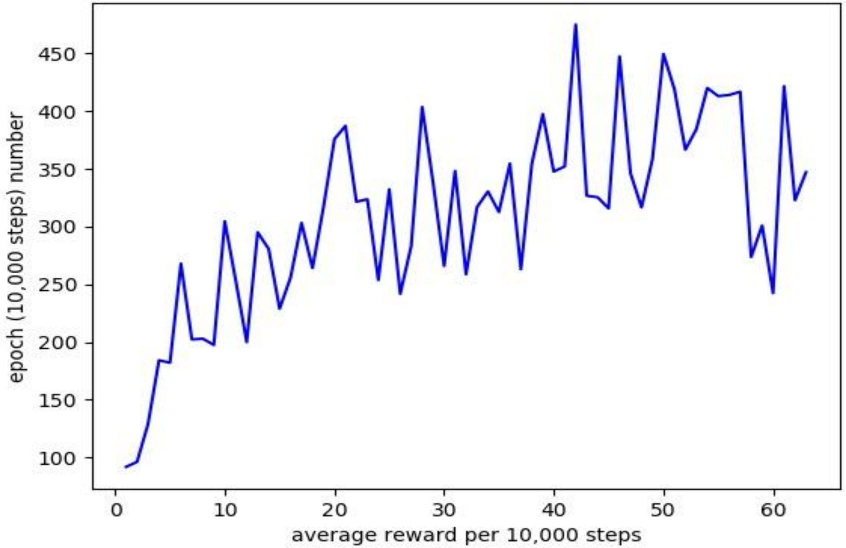
Insights:

- 2 different embeddings (VAE and random projection) did not have much of a difference



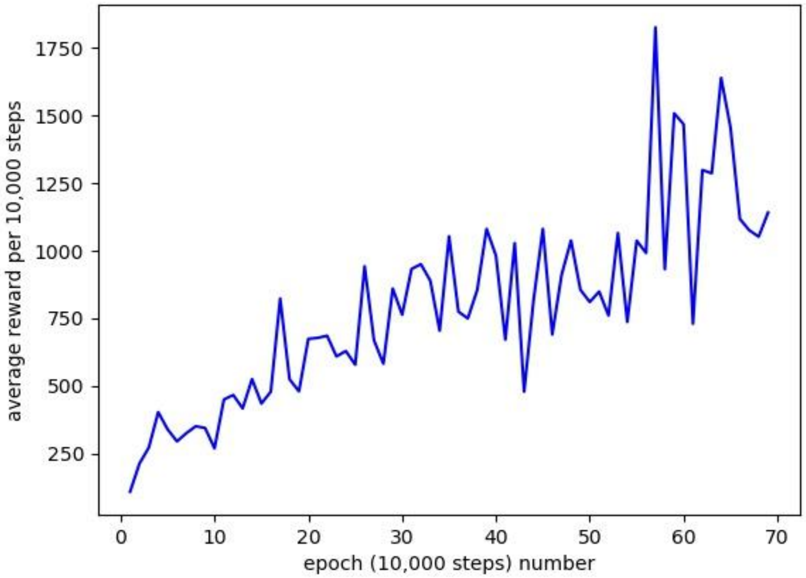
My experiments

Space invaders: K-nearest-neighbors , k = 11



My experiments

Pacman: K-nearest-neighbors , k = 11



My experiments

Space invaders: compare k = 5 and k = 11. **K = 11 (orange) seem to perform better than k = 5**