Learning to Navigate in Cities Without a Map

Piotr Mirowski, Matthew Koichi Grimes, Mateusz Malinowski, Karl Moritz Hermann, Keith Anderson, Denis Teplyashin, Karen Simonyan, Koray Kavukcuoglu, Andrew Zisserman, Raia Hadsell
DeepMind
London, United Kingdom

Presenter: MD MAHFUZ AL HASAN
Where Am I?

Where am I going?
(Deep) Reinforcement Learning

Agent \rightarrow Observations \rightarrow Rewards \rightarrow Environment

\begin{align*}
S_t, S_{t+1}, r_t \\
S_t \\
a_t
\end{align*}
Value Function

\[ V^\pi(s) = E[\sum_{t>0} \gamma^t r_t | s_0 = s, \pi ] \]

Q-Value Function

\[ Q^\pi(s,a) = E[\sum_{t>0} \gamma^t r_t | s_0 = s,a_0 = a, \pi ] \]
Reinforcement Learning with unsupervised auxiliary task.  
Jaderberg, Mnih, Czarnecki et al. (2016)
Can we solve navigation task in real world?
Street View as an RL environment: **StreetLearn**

Crop it and render at 84*84
Actions for Agent

5 Different Movements

➢ Slow Rotate Left(22.5°)
➢ Slow Rotate Right(22.5°)
➢ Fast Rotate Left(67.5°)
➢ Fast Rotate Right(67.5°)
➢ Move Forward

7000 - 65500 nodes/panorama

10 meter

3.5 to 5 km
The Courier Task

- Random start and target
- Navigation without a map
- Reward shaped when close to goal(<200m)
- Action: Rotate Left, Right or forward
- Inputs for the agent at every time point t:
  - 84*84 RGB image observations
  - Landmark based goal description
Goal Description

$$\mathcal{L} = \{(Lat^g_k, Long^g_k)\}_k$$

To represent a goal at \((Lat^g, Long^g)\)

Softmax over the distances to the \(k\) Landmarks

For distances \(\{d_{t,k}^g\}_k\) Goal Vector

$$g_{t,i} = \frac{\exp(-\alpha d_{t,i}^g)}{\sum_k \exp(-\alpha d_{t,k}^g)}$$
Architecture

Goal Nav
Agent
Architecture: Goal Navigation Agent

RGB 84 x 84

Conv. 8x8, stride 4

Conv. 4x4, stride 2

FC 256

Policy LSTM 256

$\pi(a_t)$

$V_t$

$g_t$

$a_{t-1}, r_{t-1}$
Architecture: City Nav. and Multi-city Nav. Agent

RGB 84 \times 84 \\
\rightarrow \\
\text{Conv. 8x8, stride 4} \\
\rightarrow \\
\text{ReLU} \\
\rightarrow \\
\text{Conv. 4x4, stride 2} \\
\rightarrow \\
\text{ReLU} \\
\rightarrow \\
\text{ReLU} \\
\rightarrow \\
\text{FC 256} \\
\rightarrow \\
\text{ReLU} \\
\rightarrow \\
\text{Goal LSTM 64} \\
\rightarrow \\
\theta_t \\
\rightarrow \\
\pi(a_t) \\
\rightarrow \\
\text{Policy LSTM 256} \\
\rightarrow \\
V_t \\
\rightarrow \\
a_{t-1}, r_{t-1} \\
\rightarrow \\
g_t \\
\rightarrow \\
\text{At-1, Rt-1} \\
\rightarrow \\
\text{gt}
Training

➢ Q Learning
  ○ 1 step
  ○ N step

➢ Advantage Actor-Critic
  ○ A2C
  ○ A3C (Asynchronous)
Actor-Critic

I rotate the piece

Actor

Really bad action

Critic

\[ \pi(s, a, \theta) \]

\[ \hat{q}(s, a, w) \]
$$\Delta \theta = \alpha \nabla_{\theta} (\log \pi_\theta(s,a))q_w(s,a)$$

$$\Delta w = \beta (R(s,a) + \gamma q_w(s_{t+1},a_{t+1}) - q_w(s_t,a_t)) \nabla_w q_w(s_t,a_t)$$
Advantage Actor-Critic

For each episode:

Initialize $S$, $\theta$, $w$

For timestep $t = 0,1,2,... \ldots,$max\_timestep do:

Sample $a_t \sim \pi_\theta(s_t)$

Sample $R_{t+1}$ and $S_{t+1}$

$\delta_t = R_{t+1} + \gamma v_w(s_{t+1}) - v_w(s_t)$ \quad ; Advantage Function

$w = w + \beta \delta_t \nabla_w v_w(s_t)$ \quad ; value par. update

$\theta = \theta + \alpha \delta_t \nabla_\theta (\log \pi_\theta(a_t|s_t))$ \quad ; policy par. update
Global network Parameters

Coordinator

Agent 1  Agent 2  Agent 3  ...  Agent N
Results

Average per-episode rewards (y-axis) Vs Learning steps (x-axis)

NYU (New York City)  Central London
Reward Shaping

Average per-episode rewards (y-axis) Vs Learning steps (x-axis)

\[ r_t = \max \left( 0, \min \left( 1, \frac{200 - d_t^g}{100} \right) \right) \times r^g \]
Number of steps required for the CityNav Agent to reach a goal (Y-axis) Vs Initial Straight Line Distance (X-axis)
Results: MultiCity Experiments

Transfer Learning Diagram

Transfer Learning Performance

No. of cities
Ablation Analysis

Learning Curve of City Nav Agent (2 LSTM + Skip + HD) on NYU comparing different ablations.

Learning Curve of CityNav Agent with different goal representation.
Thank You !!