10707: Deep Learning
Language Grounding and Localization

Russ Salakhutdinov

Machine Learning Department
Carnegie Mellon University
Canadian Institute for Advanced Research
Learning Behaviors

Learning to map sequences of observations to actions, for a particular goal
Reinforcement Learning

Action

$\alpha_t$

Reward

$r_t$

Observation / State

$O_t$
Deep Reinforcement RL

Action

$\alpha_t$

Reward

$r_t$

Observation / State

$O_t$
Task-oriented language grounding

Train
- Go to the short red torch
- Go to the blue keycard
- Go to the largest yellow object
- Go to the green object

Test
- Go to the tall green torch
- Go to the red keycard
- Go to the smallest blue object

Go to the green torch
Demo

https://www.youtube.com/watch?v=JziCKsLrudE
Challenges

- *recognize* objects in raw pixel input,
- *explore* the environment, handle occlusion
- *ground* each concept of the instruction in visual elements or actions,
- *reason* about the pragmatics of language, and
- *navigate* to the correct object while avoiding incorrect ones.

Single model to tackle multiple instructions
Generalize to unseen attribute-object pairs
Related work (1)

- **Grounding Language in Robotics.**

- **Mapping Instructions to Action Sequences.**
  - Chen and Mooney (2011) and Artzi and Zettlemoyer (2013): semantic parsing to map navigational instructions to a sequence of actions.
  - Mei, Bansal, and Walter (2015): neural mapping of instructions to sequence of actions
Related work (2)

- Deep reinforcement learning using visual data.
  - Deep Reinforcement learning approaches for playing FPS games (Lample and Chaplot 2016; Wu and Tian 2017; Dosovitskiy and Koltun 2017).
  - Zhu et al. (2016): target-driven visual navigation
  - Yu, Zhang, and Xu (2017): learning to navigate in a 2D maze-like environment and execute commands
  - Misra, Langford, and Artzi (2017): mapping raw visual observations and text input to actions in a 2D Blocks environment.
  - Oh et al. (2017): zero-shot task generalization in a 3D environment.
Experimental setting

Action

\[ a_t \]

State

\[ r_t \]

\[ o_t \]

Agent

Environment

Go to the green torch

State
Network overview

- **Go to the green torch**
  - Natural Language Instruction ($L$)
  - Image ($I_t$)

- **Network overview**

  - **GRU Network**
    - $g(L; \theta_{GRU})$
    - Instruction Representation
    - $x_L = g(L; \theta_{GRU})$

  - **Conv Network**
    - $f(I_t; \theta_{conv})$
    - Image Representation
    - $x_I = f(I_t; \theta_{conv})$

  - **Multimodal Fusion** ($M$)
    - State Representation
    - $M(x_L, x_I)$

  - **Policy Learning Module**
    - Policy
    - $\Pi(a | I_t, L)$
Multimodal Fusion

- Baseline Approach: Concatenation
- Proposed Approach: Gated-Attention

- Gated-Attention (Dhingra et al.)
  - attention weights for features maps, determines which filters to attend to
  - element-wise product (Gating)
  - creates instruction-specific convolutional filter representations
Gated-Attention

Image Representation
\[ x_i = f(l_i; \theta_{conv}) \]

Instruction Representation
\[ x_L = g(L; \theta_{GRU}) \]

Gated-Attention Multimodal Fusion Unit
\[ M_{GA}(x_i, x_L) = M(a_L) \odot x_i \]

Attention Vector
\[ a_L = h(x_L) \]

To policy learning module
Embodied Multimodal Learning

Dual Attention Architecture

Chaplot et al., 2019
Policy Learning

- Asynchronous Advantage Actor-Critic (A3C) (Mnih et al.)
  - uses a deep neural network to parametrize the policy and value functions and runs multiple parallel threads to update the network parameters.
  - use entropy regularization for improved exploration
  - use Generalized Advantage Estimator to reduce the variance of the policy gradient updates (Schulman et al.)
Environment

- 18 objects
- 5 types of objects
- Different colors and sizes
- Superlative instructions:
  - Largest, smallest
- Combinations
  - Tall green torch
  - Largest red object
- 70 instructions
Environment difficulty
Results
Training Progress

https://www.youtube.com/watch?v=o_G6was03N0
t-SNE Visualizations
Recent work of language grounding

- Environments
  - Home-platform [MILA, Brodeur et al. 2017]
Recent work of language grounding

- Environments
  - Home-platform [MILA, Brodeur et al. 2017]
  - House3D [FAIR, Wu et al. 2017]
Recent work of language grounding

- **Environments**
  - Home-platform [MILA, Brodeur et al. 2017]
  - House3D [FAIR, Wu et al. 2017]
  - MINOS [Intel/Princeton, Savva et al. 2017]
Recent work of language grounding

- **Environments**
  - Home-platform [MILA, Brodeur et al. 2017]
  - House3D [FAIR, Wu et al. 2017]
  - MINOS [Intel/Princeton, Savva et al. 2017]

- **Grounded Language Learning**
  [Deepmind, Hermann et al. 2017]
Recent work of language grounding

- **Environments**
  - Home-platform [MILA, Brodeur et al. 2017]
  - House3D [FAIR, Wu et al. 2017]
  - MINOS [Intel/Princeton, Savva et al. 2017]

- **Grounded Language Learning**
  [Deepmind, Hermann et al. 2017]

- **Embodied QA** [FAIR, Das et al. 2017]
Localization

Estimating the location of an autonomous agent given:

- a map of the environment
- Agent observations
Motivation

- Localization is considered as the **basic precondition for truly autonomous agents** by Burgard et al. (1998)
- Downstream tasks: exploration, target-navigation, planning
- Applications: autonomous vehicles, factory robots, housekeeping robots, delivery drones
Passive Localization
Active Localization

Agent Observations

$t = 1$

$t = 2$

Active Localization

Map Information

Predictions

Location

Action

$x$ $y$ $o$

Predictions

Location

Action

$x$ $y$ $o$
Active Localization

Agent Observations

$t = 1$

$t = 2$

Map Information

Active Localization

Predictions

$x$ $y$ $o$

Location

Action

$x$ $y$ $o$

Location

Action
Related Work

- Local Localization:
  - Kalman Filters (Smith et al., 1990)
  - Geometry-based visual odometry methods (Nister et al., 2006)
  - DeepVO (Wang et al., 2017), VINet (Clark et al., 2017)

- Global Localization:
  - Markov Localization (Fox, 1998)
  - Multi-hypothesis Kalman filters (Cox & Leonard, 1994; Roumeliotis & Bekey, 2000)
  - Monte Carlo Localization (Thrun et al., 2001)
  - Active Markov Localization (Fox et al., 1998)

- Learning policy:
  - Navigation: (Mirowski et al. 2017)
  - Planning: Value Iteration Networks (Tamar et al., 2016)
  - Planning under uncertainty: QMDP-Net (Karkus et al., 2017)
  - Mapping and Planning: Cognitive Mapper and Planner (Gupta et al., 2017)

- End-to-end Localization on known maps:
  - PoseNet (Kendall et al., 2015), VidLoc (Clark et al., 2017)
Problem Formulation

$s_t$: Agent observation at time $t$

$a_t$: Action taken by the agent at time $t$

$y_t$: Position of the agent at time $t$

$M$: Information about the map
Problem Formulation

$s_t$: Agent observation at time $t$
$a_t$: Action taken by the agent at time $t$
$y_t$: Position of the agent at time $t$
$M$: Information about the map

\[
P(y_t | s_{1:t}, a_{1:t-1}, M) : \text{Belief}
\]
\[
\pi(a_t | s_{1:t}, a_{1:t-1}, M) : \text{Policy}
\]
Bayesian Filtering

$s_t$: Agent observation at time $t$
$a_t$: Action taken by the agent at time $t$
$y_t$: Position of the agent at time $t$
$M$: Information about the map

(Fox et al., 2003)
Bayesian Filtering

$s_t$: Agent observation at time $t$
$a_t$: Action taken by the agent at time $t$
$y_t$: Position of the agent at time $t$
$M$: Information about the map

**Belief**: Probability distribution over $y_t$ conditioned over past observations $s_{1:t}$ and actions $a_{1:t-1}$:

\[
Bel(y_t) = P(y_t | s_{1:t}, a_{1:t-1}, M)
\]

**Likelihood**: Probability of observing $s_t$ given that the location of the agent is $y_t$:

\[
Lik(s_t) = P(s_t | y_t)
\]

(Fox et al., 2003)
Bayesian Filtering

$s_t$: Agent observation at time $t$

$a_t$: Action taken by the agent at time $t$

$y_t$: Position of the agent at time $t$

$M$: Information about the map

**Belief:** Probability distribution over $y_t$ conditioned over past observations $s_{1:t}$ and actions $a_{1:t-1}$:

$$Bel(y_t) = P(y_t | s_{1:t}, a_{1:t-1}, M)$$

**Likelihood:** Probability of observing $s_t$ given that the location of the agent is $y_t$:

$$Lik(s_t) = P(s_t | y_t)$$

**Under the Markov assumption:**

$$Bel(y_t) = \sum_{y_{t-1}} P(y_t | y_{t-1}, a_{t-1}) Bel(y_{t-1})$$

Belief before observing $s_t$

Transition function

Belief after observing $s_{t-1}$

$$Bel(y_t) = \frac{1}{Z} Lik(s_t) Bel(y_t)$$

Belief after observing $s_t$

Prob. of observing $s_t$

Belief before observing $s_t$

**Transition function:** Probability of landing in a state $y_t$ from $y_{t-1}$, based on the action, $a_{t-1}$:

$$f_T = P(y_t | y_{t-1}, a_{t-1})$$

(Fox et al., 2003)
Bayesian Filtering

\( s_t \): Agent observation at time \( t \)
\( a_t \): Action taken by the agent at time \( t \)
\( y_t \): Position of the agent at time \( t \)
\( M \): Information about the map

**Belief:** Probability distribution over \( y_t \) conditioned over past observations \( s_{1:t} \) and actions \( a_{1:t-1} \):

\[
Bel(y_t) = P(y_t | s_{1:t}, a_{1:t-1}, M)
\]

**Likelihood:** Probability of observing \( s_t \) given that the location of the agent is \( y_t \):

\[
Lik(s_t) = P(s_t | y_t)
\]

Under the Markov assumption:

\[
\overline{Bel}(y_t) = \sum_{y_{t-1}} P(y_{t-1}, a_{t-1}) Bel(y_{t-1})
\]

Belief before observing \( s_t \)
Belief after observing \( y_{t-1} \)
Transition function
Belief after observing \( s_{t-1} \)

\[
Bel(y_t) = \frac{1}{Z} Lik(s_t) \overline{Bel}(y_t)
\]

Belief after observing \( s_t \)
Belief before observing \( s_t \)
Prob. of observing \( s_t \)

(Fox et al., 2003)
Bayesian Filtering

$s_t$: Agent observation at time $t$  
$a_t$: Action taken by the agent at time $t$  
$y_t$: Position of the agent at time $t$  
$M$: Information about the map

**Belief:** Probability distribution over $y_t$ conditioned over past observations $s_{1:t}$ and actions $a_{1:t-1}$:

$$Bel(y_t) = P(y_t | s_{1:t}, a_{1:t-1}, M)$$

**Likelihood:** Probability of observing $s_t$ given that the location of the agent is $y_t$:

$$Lik(s_t) = P(s_t | y_t)$$

Under the Markov assumption:

$$\overline{Bel}(y_t) = \sum_{y_{t-1}} P(y_t | y_{t-1}, a_{t-1}) Bel(y_{t-1})$$

- Belief before observing $s_t$  
- Transition function  
- Belief after observing $s_{t-1}$

$$Bel(y_t) = \frac{1}{Z} Lik(s_t) \overline{Bel}(y_t)$$

- Belief after observing $s_t$  
- Prob. of observing $s_t$  
- Belief before observing $s_t$

(Fox et al., 2003)
Representation of Belief and Likelihood

\[ x \rightarrow y \rightarrow \theta \]

Map size
Number of orientations

\[ O \times M \times N \]
Representation of Belief and Likelihood

3-dimensional tensor representing $x$-coordinate, $y$-coordinate and orientation

$O \times M \times N$

Map size

Number of orientations
Representation of Belief and Likelihood

3-dimensional tensor representing $x$-coordinate, $y$-coordinate and orientation

Each element represents the probability of the agent being present in the corresponding location

$O \times M \times N$

Map size

Number of orientations
Simulation Environments

Top View | Agent’s Observation | Likelihood Map | Map Design
---|---|---|---
Maze3D | | | 
Unreal3D | | | 

East | North | West | South
Active Neural Localization
Active Neural Localization

Belief before observing $s_t (\overline{Bel}(y_t))$

<table>
<thead>
<tr>
<th>East</th>
<th>North</th>
<th>West</th>
<th>South</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$\overline{Bel}(y_t)$
Active Neural Localization

Agent’s observation \((s_t)\)  

Belief before observing \(s_t (\overline{Bel}(y_t))\)

\(t = 1\)

\(s_1\)

Perceptual Model

\(Lik(s_1)\)
Active Neural Localization

Agent's observation ($s_t$)  Belief before observing $s_t$ ($\overline{Bel}(y_t)$)

$t = 1$

Perceptual Model

Belief after observing ($\overline{Bel}(y_t)$)

Policy Model

Feature Representation

Conv1
- 32 filters
- $8 \times 8$
- stride 4

Conv2
- 64 filters
- $4 \times 4$
- stride 2

Fully-Connected
- 512

Map Design & agent's true location
Active Neural Localization

Agent’s observation ($s_t$)  Belief before observing $s_t$ ($\overline{Bel}(y_t)$)

Perceptual Model: $\overline{Bel}(y_t)$

Belief after observing $s_t$ (Lik($s_1$))
Active Neural Localization

Agent's observation \( s_t \)

Belief before observing \( s_t \) \((\overline{\text{Bel}}(y_t))\)

Belief after observing \( s_t \) \((\text{Bel}(y_t))\)

\[
\text{Bel}(y_t) = \frac{1}{Z} \text{Lik}(s_t) \overline{\text{Bel}}(y_t)
\]

Belief after observing \( s_t \)

Belief before observing \( s_t \)
Active Neural Localization

Agent's observation ($s_t$)

Belief before observing $s_t$ ($\overline{Bel}(y_t)$)

Perceptual Model

Belief after observing $s_t$ ($Bel(y_t)$)

$t = 1$
Active Neural Localization

Agent's observation ($s_t$) → Belief before observing $s_t$ ($\text{Bel}(y_t)$) → Belief after observing $s_t$ ($\text{Bel}(y_t)$)

Perceptual Model

Map Design & agent's true location

$t = 1$, $s_1$
Active Neural Localization

Agent's observation ($s_t$) | Belief before observing $s_t (\overline{Bel}(y_t))$ | Belief after observing $s_t (Bel(y_t))$
---|---|---
$t = 1$ | | |
$\overline{Bel}(y_1)$ | Lik($s_1$) | $Bel(y_1)$
East | North | West | South | East | North | West | South
---|---|---|---|---|---|---|---|---
Active Neural Localization

Agent's observation \((s_t)\)  
Belief before observing \(s_t (\overline{Bel}(y_t))\)  

\[
\begin{array}{c|c|c|c}
\text{East} & \text{North} & \text{West} & \text{South} \\
\hline
\text{Bel}(y_1) & & & \\
\hline
\text{Lik}(s_1) & & & \\
\end{array}
\]

Belief after observing \(s_t (Bel(y_t))\)  

\[
\begin{array}{c|c|c|c}
\text{East} & \text{North} & \text{West} & \text{South} \\
\hline
\\text{Bel}(y_1) & & & \\
\hline
\end{array}
\]

Policy Model  

\(a_1 = '\text{Turn left}'\)
Active Neural Localization

Agent's observation ($s_t$)  Belief before observing $s_t$ ($\text{Bel}(y_t)$)  Belief after observing $s_t$ ($\text{Bel}(y_t)$)

Perceptual Model  $\text{Lik}(s_1)$  Policy Model

$t = 1$

$\text{Bel}(y_1)$

East  North  West  South

$\text{Bel}(y_1)$

East  North  West  South

$a_1 = '\text{Turn left}'$

Policy Model

Acton Layer (FC)  Critic Layer (FC)  Value

256  40  8

Fully-Connected

Embedding

Map Design  &  agent's true location  Agent's perspective

Belief $O \times M \times N$  Conv1  Conv2  Flatten

16 filters  7 x 7  stride 3  16 filters  3 x 3  stride 1

16 filters  3 x 3  stride 1

Action History (5 actions)  Timestep

Map Design & agent's true location  Agent's perspective

Agent's perspective

North  East  South  West
Active Neural Localization

Agent's observation ($s_t$)  Belief before observing $s_t$ ($\overline{Bel}(y_t)$)  Belief after observing $s_t$ ($Bel(y_t)$)  Map Design & agent's true location  Agent's perspective

$\text{Perceptual Model}$  $Lik(s_1)$  $\text{Policy Model}$

$\text{Bel}(y_1)$  $f_T$  $a_1 = \text{`Turn left' }$

$t = 1$  $s_1$  $\text{East}$  $\text{North}$  $\text{West}$  $\text{South}$  $\text{East}$  $\text{North}$  $\text{West}$  $\text{South}$  $\text{East}$  $\text{North}$  $\text{West}$  $\text{South}$
Active Neural Localization

Belief before observing $s_t$ ($\overline{Bel}(y_t)$)

Belief after observing $s_t$ ($Bel(y_t)$)

Belief before observing $s_t-1$ ($\overline{Bel}(y_{t-1})$)

Transition function

Belief after observing $s_{t-1}$

Belief before observing $s_t$ ($\overline{Bel}(y_t)$) = $\sum_{y_{t-1}} P(y_t|y_{t-1}, a_{t-1}) \overline{Bel}(y_{t-1})$
Active Neural Localization

Agent's observation ($s_t$)

Belief before observing $s_t$ ($\bar{Bel}(y_t)$)

Belief after observing $s_t$ ($Bel(y_t)$)

Perceptual Model

Lik($s_1$)

Lik($s_2$)

Policy Model

Agent's perspective

Map Design & agent's true location
Active Neural Localization

Perceptual Model

$t = 2$

$s_2$

Lik($s_2$)

$t = 3$

$s_3$

Lik($s_3$)

Policy Model

$Bel(y_2)$

$Bel(y_3)$

$a_2 = \text{"Turn left"}$

$a_3 = \text{"Forward"}$
Active Neural Localization

Perceptual Model

\[ \text{Lik}(s_t) \]

Policy Model

\[ f_T \]

Agent’s observation

\[ \text{Bel}(y_t) \]

\[ a_t = \text{Forward} \]

Map Design & agent’s true location

\[ Q^T = \text{Forward} \]

\[ Q^V = \text{Forward} \]

\[ Q^Z = \text{Turn left} \]

\[ K = 1, 2, 3, 4, 5, 6 \]
Optimization

- At the end of the episode, the location prediction is the element with the maximum probability in the belief tensor.
- The agent receives a positive reward (+1) for correct prediction.
- The entire model is trained end-to-end with reinforcement learning, specifically Asynchronous Advantage Actor-Critic (A3C).
Example

\[ Bel(y_4) \]

\[ t=4 \]

East  North  West  South
Example

\[ Bel(y_4) \]

\[ t=4 \]
Example

Bel(y₄)

East  North  West  South

t=4
Example

$$Bel(y_5)$$

$t=5$

East North West South
Example

\[ Bel(y_6) \]

\[ t=6 \]

Diagram showing four directions (East, North, West, South) with a belief distribution. The agent's belief over the location is visualized in the diagram on the right.
Experiments

Unseen Mazes

Unseen Textures

Dynamic Lighting

Domain Adaptation
Demo video: Doom

https://www.youtube.com/watch?v=rdhKu8GqVLw
Demo video: Unreal

https://www.youtube.com/watch?v=T5Ezx-_QfU0
Pose Estimation: Towards Deep SLAM

Parisotto et al., CVPR Workshop on Visual SLAM 2018
Building Intelligent Agents

Navigate Autonomously
Localize and Plan
Multi-modal Input
Perceptive Human Speech
Reason & Understand Language
Recognize objects
Thank you