CS231N

Convolutional Neural Networks for Visual Recognition

Spring 2017

Target-Driven Navigation with Imitation Learning

Jongho Kim, Seungbin Jeong

Motivation

- Robotics problems in general involve the agents' interactions with physical environments and objects.
- Typical deep reinforcement learning (DRL) based methods (e.g. a policy gradient RL for locomotion of a four-legged robot [1], deep Q-networks on ATARI games) have shown promising results.
- However, there are some well know issues of DRL such as
  - specification of suitable reward function for goal achievement
  - training time complexity (requires several costly episodes of trial-and-error to converge)
- The concept of imitation learning arises: supervised learning assists the agent to mimic an expert policy without usage of reward function for faster convergence.
- We focus our work on the problem of navigating a space to find a given target using only image inputs. (Without the map data.)

Problem Statement

- **GOAL**: Find the *minimal* sequence of actions that lead the agent from current location to a target.
- **Model Output**: an action in 3D ($i.e.$, move forward)
  - model learns a mapping from 2D image to an action in 3D space

Target-driven Navigation

- Standard DRL models: **Input**: current state, s_t
  Find a stochastic policy function π that maps a given state into an action, $a \sim \pi(s_t)$.
  - Requires re-training for each different target in the environment with multiple navigation targets.
  - A lack of generalization (re-train new models for new targets)
- Target-driven Navigation: **Input**: current state, s_t at time t, the target g.
  Learns a stochastic policy function π which takes s_t and g as inputs and maps them into an action. ($i, e.$, learn the model parameters θ with DRL), $a \sim \pi(s_t, g \mid \theta)$

Living room scene — Kitchen scene — Bathroom scene

Figure 1: **The AI2 The House Of inteRactions (THOR) Framework**

The AI2-THOR Framework

- Need a framework for performing actions and receiving their outcomes in 3D environment.
- Designed by integrating a physics engine (*Unity 3D*) with a deep learning framework (*Tensorflow*).
- Photo-realistic and available for direct communication which allows instant feedback (causality) from environment used for online decision making. Detailed discussion on this framework can be found in [2]

Learning Setup

- *Action space*: Discretized four actions (move forward, turn right, turn left and move backward). Used constant step length (0.5 meters) and turning angle of 90 degree.
  - To model uncertainty, added a Gaussian noise to steps $\mathcal{N}(0, 0.01)$ and turns $\mathcal{N}(0, 1.0)$
- *Observation*: Both the current sight and target are images taken by the agent's RGB camera in its first-person view.

Imitation Learning Model

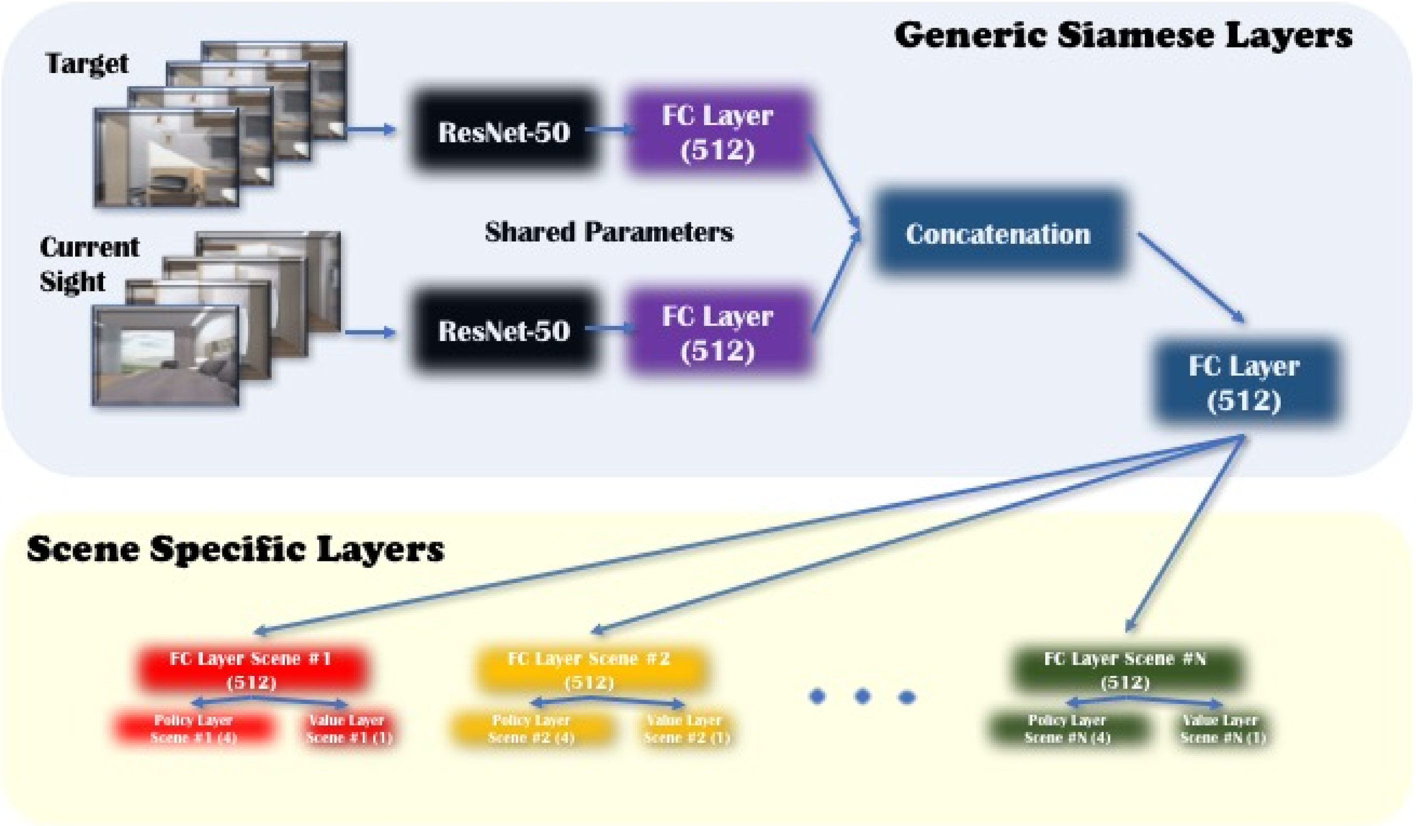

Figure 2: **Network architecture of our imitation learning model**

- Generated expert policy (ground truth) using shortest path algorithm, $a_E \sim \pi_E(s_t, g)$.
- The ResNet-50 layers are pre-trained on ImageNet. It produces **2048-d** features on a 224x224x3 RGB image (we truncated the softmax layer) and its parameters are fixed during training.
- To address agent's past actions, we concatenate features of 4 recent history frames.
- Activations from both current observation/target sides are concatenated. (becomes **1024-d** space)
- These **8192-d** output vectors (one from current state and one from target) are passed to a fully-connected (FC) layer and projected into **512-d** space.
- The projected representation is passed through *scene-specific* layers to learn unique characteristics of a scene that are important for navigation tasks. scene-specific layers produces 4 policy outputs.
- Using the expert policy, we train this network with a shared ADAM optimizer of learning rate 4×10^{-3} with decay rate 0.9. We used l2-regularization with regularization constant 5×10^{-5}.

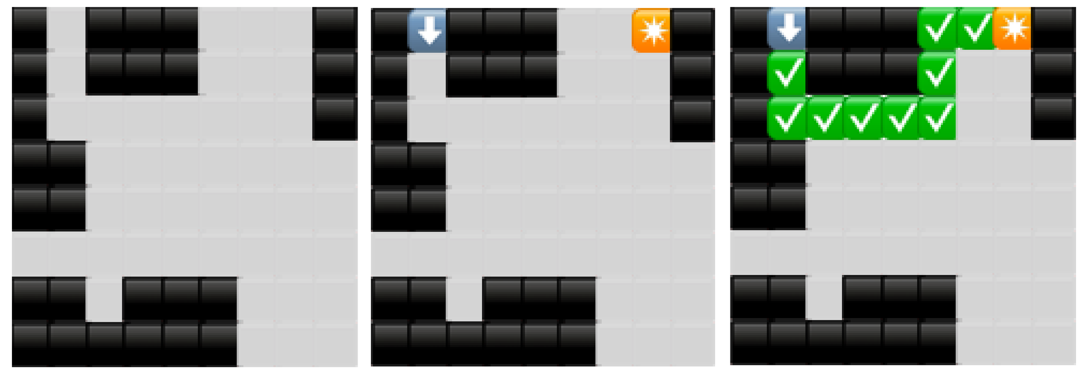

Figure 3: Shortest path search between current state and target

Experiment

	Maximum Steps	Minimum Steps	Average Steps	Standard Deviation
DRL 1 hour	100.0	100.0	100.0	100.0
IL 1 hour	239.6	38.4	282.2	447.7
DRL 2 hours	115.9	81.6	98.5	147.9
IL 2 hours	135.6	20.1	71.6	203.6
DRL 1 hour+ IL 1 hour	274.5	24.4	290.2	609.9
IL 1 hour+ DRL 1 hour	93.2	80.6	112.8	99.2

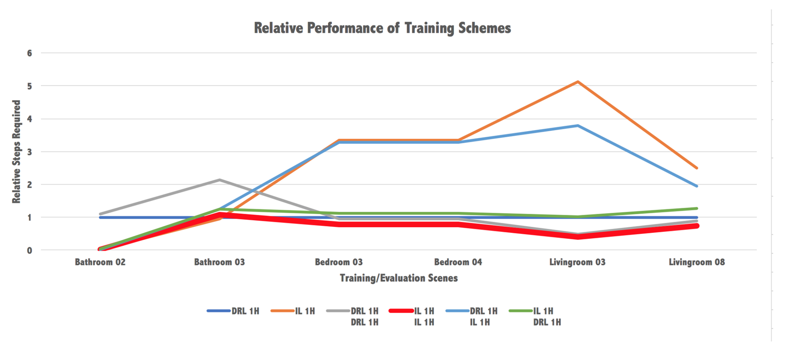

Figure 4: **Target Generalization.**

Imitation Learning.

- For training sets, we chose two different types: optimal-path history/target pairs and sole current sight observation/target pairs.
  - The training sets with optimal-path history/target pairs are expected to encourage the agent to follow optimal paths once it enters the path.
  - The training sets with sole current observation/target pairs emphasizes the importance of current observation rather than past observation memories.

Imitation Learning with Reinforcement Learning.

- We expected the imitation learning can be combined with conventional reinforcement learning to have synergetic effects.

Results.

- Figure 4 shows the result of the target generalization experiment.
  - The numbers in the table are equivalent number of steps for each training method to 100 steps of 1-hour DRL-trained model. Similarly, the number in the plot shows required steps of each training results equivalent to 1 step of 1-hour DRL-trained model.
  - For a short-term training, the imitation learning shows unsatisfactory performance in most aspects.
  - But after a 2-hour training, it outperforms any other combination of DRL or IL. (28.4% decrease in required steps.)
  - Discouragingly, the combination of two training method did not show synergetic effects. Doing preliminary imitation learning hardly had any effects. Post-imitation-learning on DRL-training model rather deteriorated the performance.

Conclusion/Future Directions

- Our results lead to many extensions of this work. For example, instead of supervised learning (behavioral cloning) approach, we could use *inverse reinforcement learning* (IRL) to obtain a suitable reward function for navigation tasks.
- We could evaluate our model in environments with dynamic changes or perhaps longer distances and to build models suitable for learning the physical interactions and objects in the framework.

References.

[1] N. Kohl and P. Stone. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2004.

[2] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. *CoRR*, abs/1609.05143, 2016.