Review2: Learning to Navigate in Complex Environments

Vivswan Shitole

CS539 Embodied AI

1 Summary

In this paper, the authors propose that the goal driven reinforcement learning formulation of the navigation problem can be dramatically improved in terms of data efficiency and task performance by jointly learning additional auxiliary tasks that leverage multi-modal sensory input. The authors hypothesize that an RL agent learning a policy that maximizes reward can learn navigational abilities as a by-product of minimizing augmented loss of the auxiliary tasks. This intrinsic end-to-end training approach results in learning a policy that is derived from representation consisting of task relevant features. Augmented loss of auxiliary tasks provide dense training signals that can complement sparse rewards. More concretely, the stated approach uses depth prediction and SLAM-inspired loop closure classification as the auxiliary tasks. The RL problem of maximizing cumulative reward is addressed with Asynchronous Actor Critic (A3C) algorithm. Both the policy and the value function share all intermediate representations, both being computed using a separate linear layer from the topmost layer of the model. The paper introduces four different architectures of the network model based on different combinations of representation network and auxiliary targets: (a) **FF A3C**: representation network is a convolutional encoder followed by a feed-forward layer for policy and value function outputs. (b) **LSTM A3C**: has an LSTM layer over the convolutional encoder in FF A3C (to address memory requirements of the tasks). (c) **Nav A3C**: has stacked LSTM layers over the convolutional encoder (as memory over different time scales) and uses additional inputs for reward, previous action and agent-relative velocity (velocity is integrated to predict loop closure). (d) **Nav A3C + D1D2L**: similar to Nav A3C but has additional outputs to predict depth (D1 if depth predicted from the convolutional layer, D2 if depth predicted from the top LSTM layer) and loop closure (L). The auxiliary losses are computed on the current frame via a single layer MLP. The agent is trained by applying a weighted sum of gradients coming from A3C, the gradients from depth prediction and the gradients from loop closure.
Experiments are conducted using first-person 3D mazes from the DeepMind Lab environment. The aforementioned different agent architectures are evaluated by training on 5 different mazes: a small and a large static maze (fixed goal and fruit locations), a small and a large dynamic maze (fruits and goals are randomly placed on every episode) and a dynamic I-maze. The mean over top 5 runs as well as the top 5 curves are plotted as learning curves. The Nav A3C + D2 agent yields the best performance curves in most cases. To validate that the agents implicitly learn an internal representation of location for navigation, the authors train a position decoder that takes the internal representation (hidden state of LSTM or last-layer features of conv net) as input and outputs a multinomial probability distribution over the discretized maze locations. These probability maps show that the agent has initial uncertainty on location at the start of an episode, which later gives way to accurate position prediction as the agent navigates. The authors analyze the policies learnt by stacked LSTM networks by applying tSNE dimension reduction to the cell activations at each step for each of the four goal locations in the I-maze. While clusters corresponding to the LSTM A3C agent are distinct for each of the 4 goals, Nav A3C agent yields 2 main clusters, suggesting that the LSTM representing the policy for the Nav A3C agent maintains an efficient representation of 2 sub-policies (for the two arms of the I-maze). Finally the authors conduct a comparative study of the different agent architectures, providing tabular results for AUC, score, goal-hitting percentage, position estimation accuracy and latency of the agents over the 5 mazes.

2 Strengths

The strength of this paper is that the authors provide explanation for all of the nuances in their approach and validate them using experiments. In addition to the performance curves which validate the main hypothesis that auxiliary tasks benefit performance, the authors provide diagrams representing examples of depth predictions, loop closures

(with false positives and false negatives), policy cluster representations and position probability predictions. These diagrams validate the learning of task specific features. The authors argue that depth prediction should be incorporated as an auxiliary loss rather than a direct input as it would help build useful features for the RL problem. The authors validate this argument comparing agents with depth as input in one case and as an additional loss in the other. The authors explore two variants of the depth prediction loss - regression loss and classification loss. The authors argue that mean square error based regression loss imposes a unimodal distribution, whereas, classification loss imposes a richer non-uniform distribution over different discretized bands of depth. The authors validate this argument providing performance curves for both the loss formulations. The authors motivate the use of memory (stacked LSTM) in dynamic environments by pointing out that the for static mazes, both feedforward A3C (memoryless) and its LSTM version (with memory) perform at par.

3 Weaknesses

The experimental environment uses intermediate rewards (fruits) in addition to the final reward in order to encourage exploration. These intermediate rewards should not be needed if the agent learns a good exploration strategy using only the sparse final rewards. The auxiliary task losses provide a denser training signal anyway. The use of I-maze environment by the authors to validate the arguments for LSTM based policy representation seems like an experiment particularly designed to support the case. Another unvalidated argument is feeding the velocity and action vector to the second LSTM layer, with the first LSTM layer recieving only the reward. The authors postulate that the first layer might be able to make associations between reward and visual observations that are provided as context to the second layer from which the policy is computed, but never validate this using an experiment.

4 Reflections

This paper is a great addition to the literature on solving the navigation problem. The paper provides a strong case for using deep-RL based approach by justifying its ability to implicitly learn navigation abilities when jointly trained using auxiliary tasks that promote learning task relevant features. The paper introduces an approach that stands comparable and in close competition to SLAM. Future research comparing the two approaches would be interesting.

5 Most Interesting Thought

This paper is one more evidence to the repeated pattern observed in deep learning that deep models, be it CNN, LSTM or their combinations, perform the best when they are jointly trained over the specific task in an end-to-end manner. They implicitly learn task specific features which are rich enough to perform as good as possible on the task at hand. But these features may be too specific to the task at hand that they don't readily generalize to other task. Transfer learning in RL is still a dream.