Introduction

Learning to play text games is an important task in the language understanding domain. Text-based games are structured as a “dialog” between a game engine and the player.

TextWorld by Microsoft Research - a learning environment for training and evaluation of Reinforcement Learning agents on text-based games

First TextWorld Problems: A Reinforcement and Language Learning Challenge - a ML competition in the TextWorld framework

First TextWorld Problems competition

- An agent acts within the house.
- It tries to gather the ingredients in order to cook a delicious meal.
- The agent must:
  - figure out the ingredients from a recipe.
  - explore the house to collect them.
  - once done, go to the kitchen to do the cooking.

Data

4440 different games: each game is a data file for the TextWorld framework with the full description of the world.

The games in the training set have a different complexity, which is determined by the “skills” that the agent needs to learn in order to perform well in that game.

The skills include the number of ingredients in the recipe, actions (open, cook, cut, and “drop”) and the number of locations in the game.

Handicap—an agent could request additional information for playing the games at the cost of a score penalty.

Challenges

For AI agent to play IF games efficiently it needs to:

- master language understanding
- deal with a combinatorial action space
- perform efficient exploration
- have memory and capability for sequential decision-making

Most of the commercially available text-based games are beyond capabilities of the existing algorithmic approaches.

Approach

Game representation: sequential decision making problem, where at each time step the agent receives strings of text with the description of the current state and several strings with all possible actions.

The agent chooses one of these actions with the goal of maximizing future rewards.

Q-Learning

- A discounted reward: $R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots$
- Q-function - the expected reward when following a policy $\pi(a | s)$
- Q-learning - an algorithm to find optimal Q-function: $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \eta \cdot (r_t + \gamma \cdot \max Q(s_{t+1}, a) - Q(s_t, a_t))$

Model Architecture

Based on DRRN

Uses trainable, randomly initialized, embeddings

Adds prioritized buffer for more valuable or recent experiences

Uses RNN as a state encoder

GRU as a recurrent unit

DRRN architecture

- Uses 2 separate embeddings for states and actions
- Applies feed-forward networks to the BOW of embeddings to obtain Q-values
- The final Q-value is a dot product of states and actions Q-values

Results

<table>
<thead>
<tr>
<th>Model architecture</th>
<th>Easy (S)</th>
<th>Moderate (G)</th>
<th>Hard (H)</th>
<th>Episodes</th>
<th>Time, sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random baseline</td>
<td>1.2</td>
<td>0.8</td>
<td>0.3</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>DRRN-BASIC</td>
<td>3.0</td>
<td>2.5</td>
<td>1.5</td>
<td>120</td>
<td>362</td>
</tr>
<tr>
<td>DRRN-PROR</td>
<td>3.0</td>
<td>5.2</td>
<td>2.5</td>
<td>52</td>
<td>156</td>
</tr>
<tr>
<td>DRRN-GRU-BOTH</td>
<td>3.0</td>
<td>4.5</td>
<td>4.2</td>
<td>15</td>
<td>297</td>
</tr>
<tr>
<td>DRRN-SHARED-EMB</td>
<td>3.0</td>
<td>6.0</td>
<td>5.3</td>
<td>27</td>
<td>80</td>
</tr>
<tr>
<td>DRRN-GRU-STATE</td>
<td>3.0</td>
<td>6.0</td>
<td>8.0</td>
<td>8</td>
<td>245</td>
</tr>
</tbody>
</table>

Future Work

Generalize to unseen games

Use target network in DQN

Reduce handicap

Learn epsilon in ε-greedy

Hyper parameters

<table>
<thead>
<tr>
<th>Hyper Parameter</th>
<th>Value</th>
<th>Search Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimizer</td>
<td>Adam</td>
<td>[Adam, RMSProp]</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.001</td>
<td>[0.1 ... 0.00001]</td>
</tr>
<tr>
<td>Dropout</td>
<td>0.9</td>
<td>[0.5 ... 1.0]</td>
</tr>
<tr>
<td>Gamma</td>
<td>0.5</td>
<td>[0.4 ... 1.0]</td>
</tr>
<tr>
<td>RNN encoder</td>
<td>GRU</td>
<td>[GRU, LSTM]</td>
</tr>
<tr>
<td>RNN hidden size</td>
<td>128</td>
<td>[32 ... 256]</td>
</tr>
<tr>
<td>Grad norm</td>
<td>5.0</td>
<td>[None: 1.0 ... 10.0]</td>
</tr>
</tbody>
</table>