Recitation 11:

Quiz 3 Review 1

Alex Singh and Robin Schmucker

Overview: Topics Quiz 3

Topics:

- All papers not marked “optional”
- All topics covered in Quiz 1 and Quiz 2
- MuZero
- MBRL (LQR, iLQR, MPC, Dreamer, …)
- Intelligent Exploration
- Offline RL
- Sim2Real
- Visual Imitation Learning
- Self-supervised Visual Learning

Quiz Date: Tuesday 05/03
most likely at 5:30pm

Plan for Today:

- Recap MCTS
- Recap LSTMs
- Discussion Quiz 1
- Discussion Quiz 2

Feel free to ask any questions you have about class :-)

MCTS in MuZero

- Every node in tree associated with state *s* (Node class in code)
- For every action *a*, we have an edge *(s,a)* storing the following:
  - N(s,a) - Visit counts
  - Q(s,a) - Estimated mean Q-value
  - P(s,a) - Policy prior given by network
  - R(s, a) - Reward
  - S(s, a) - State Transition

Action Selection in MCTS Tree

- N(s,b) = parent visit count
- Leaf Node is an unexpanded node
- What does each term here do?

Selection: Each simulation starts from the internal root state s^0, and finishes when the simulation reaches a leaf node s^l. For each hypothetical time-step $k = 1...l$ of the simulation, an action a^k is selected according to the stored statistics for internal state s^{k-1}, by maximizing over an upper confidence bound [32][39],

$$a^k = \arg\max_a \left[Q(s,a) + P(s,a) \cdot \frac{\sqrt{\sum_b N(s,b)}}{1 + N(s,a)} \left(c_1 + \log\left(\frac{\sum_b N(s,b) + c_2 + 1}{c_2} \right) \right) \right] \tag{2}$$

The constants c_1 and c_2 are used to control the influence of the prior $P(s,a)$ relative to the value $Q(s,a)$ as nodes are visited more often. In our experiments, $c_1 = 1.25$ and $c_2 = 19652$.

For $k < l$, the next state and reward are looked up in the state transition and reward table $s^k = S(s^{k-1}, a^k)$, $r^k = R(s^{k-1}, a^k)$.

Node Expansion in MCTS Tree

- When we reach a leaf (unexpanded) node s_l , we:
    - Compute reward, policy and value from network (either initial or recurrent models)
- Then, initialize each edge (s_l, a) as follows:
    - Q-value as 0 (this is a design choice)
    - Visit count to 0
    - Policy prior to prior over action from policy

Backup Step in MCTS Tree

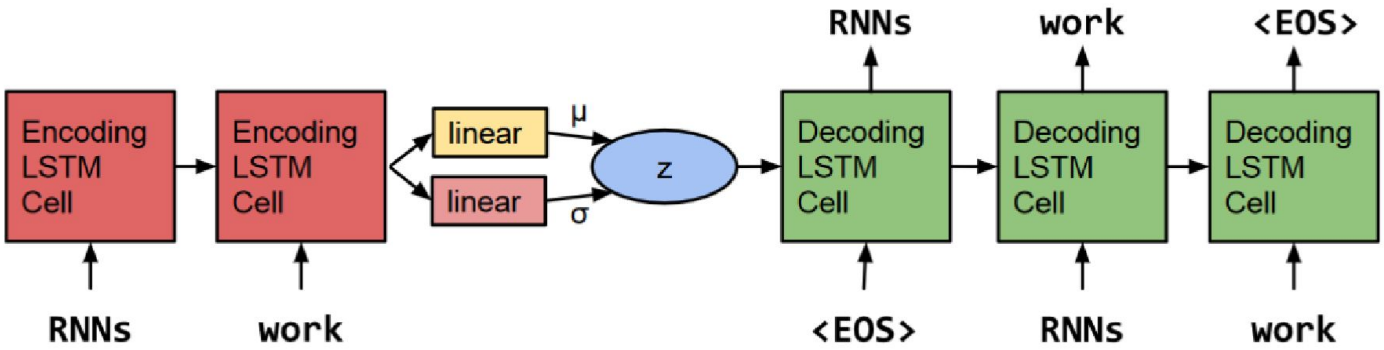
Backup: At the end of the simulation, the statistics along the trajectory are updated. The backup is generalized to the case where the environment can emit intermediate rewards, have a discount γ different from 1, and the value estimates are unbounded [3]. For $k = l...0$, we form an $l - k$-step estimate of the cumulative discounted reward, bootstrapping from the value function v^l,

$$G^k = \sum_{\tau=0}^{l-1-k} \gamma^\tau r_{k+1+\tau} + \gamma^{l-k} v^l \tag{3}$$

For $k = l...1$, we update the statistics for each edge (s^{k-1}, a^k) in the simulation path as follows,

$$\begin{aligned} Q(s^{k-1}, a^k) &:= \frac{N(s^{k-1}, a^k) \cdot Q(s^{k-1}, a^k) + G^k}{N(s^{k-1}, a^k) + 1} \\ N(s^{k-1}, a^k) &:= N(s^{k-1}, a^k) + 1 \end{aligned} \tag{4}$$

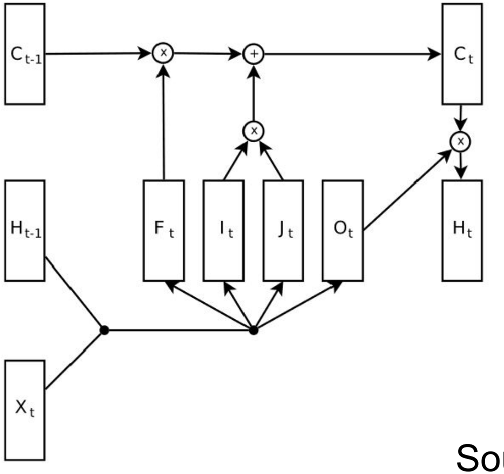
Long-Short Term Memory (LSTM



Source: Bowman

- Popular for sequential input data (text, audio, ...)
- Processes sequence one token at a time
- Maintains an internal state over time to capture long-term dependencies in the sequence

Long-Short Term Memory (LSTM



$$
\begin{aligned}
i_t &= \tanh(W_{\mathrm{xi}} x_t + W_{\mathrm{hi}} h_{t-1} + b_{\mathrm{i}}) \\
j_t &= \mathrm{sigm}(W_{\mathrm{xj}} x_t + W_{\mathrm{hj}} h_{t-1} + b_{\mathrm{j}}) \\
f_t &= \mathrm{sigm}(W_{\mathrm{xf}} x_t + W_{\mathrm{hf}} h_{t-1} + b_{\mathrm{f}}) \\
o_t &= \tanh(W_{\mathrm{xo}} x_t + W_{\mathrm{ho}} h_{t-1} + b_{\mathrm{o}}) \\
c_t &= c_{t-1} \odot f_t + i_t \odot j_t \\
h_t &= \tanh(c_t) \odot o_t
\end{aligned}
$$

Source: [Jozefowicz]

- Uses input-, output- and forget-gates to regulate the update of the cell- and hidden-state.
- Problem: To determine the RNN gradient one needs to “unroll” the prediction sequence -> slow to train

Long-Short Term Memory (LSTM

Some references:

- Brief overview: Blog by Olah
- Detailed walkthrough: Tutorial by Staudemeyer and Morris
- In practice: Tutorial by PyTorch