Sequence Tutor: Conservative Fine-Tuning of Sequence Generation Models with KL-control

Wei Bi

Tencent AI Lab

2018 June 26

Sequence Tutor: Conservative Fine-Tuning of Sequence Generation Models with KL-control

This paper proposes a general method for improving the structure and quality of sequences generated by Seq2seq.
To apply RL to sequence generation:

- Generating the next token in the sequence is treated as an action a.
- The state of the environment consists of all of the tokens generated so far, i.e. $s_t = \{a_1, a_2, \ldots, a_{t-1}\}$
- Given action a_t, we would like the reward r_t to combine information about the prior policy $p(a_t|s_t)$ as output by the Reward RNN, as well as some domain- or task-specific rewards r_T.

DQN

Given the state of the environment at time t, s_t, the agent takes an action at according to its policy $\pi(a_t|s_t)$, receives a reward $r(s_t, a_t)$, and the environment transitions to state, s_{t+1}. The optimal deterministic policy π^* satisfies the Bellman optimality equation

$$Q(s_t, a_t, \pi^*) = r(s_t, a_t) + \gamma E_{p(s_{t+1}|s_t, a_t)}[max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \pi^*)]$$

DQN approximates $Q(s, q; \theta)$ by a DNN:

$$L(\theta) = E_\beta[(r(s, a) + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta))^2]$$

- β is the exploration policy.
- θ^- is the parameters of the target Q-network that is held fixed during the gradient computation.

Sequence Tutor

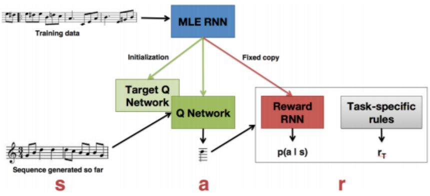

Figure 1: An RNN pre-trained on data using MLE supplies the initial weights for the Q-network and target Q-network, and a fixed copy is used as the Reward RNN.

- Pretrain a Seq2seq and fix it as a Reward RNN.
- Copy the pretrained Seq2seq network as the Target Q Network and Q network for the DQN learning.
- The reward at time t: $r(s, a) = \log p(a|s) + r_T(a, s)/c$.
- The ojbective and learned policy of DQN:

$$L(\theta) = E_\beta[\log p(a|s) + r_T(a, s)/c + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta))^2]$$

$$\pi_\theta(a|s) = \delta(a = \arg\max Q(s, a; \theta))$$

Sequence Tutor...

- DQN learns a deterministic policy, not be ideal for sequence generation.
- The problem can be expressed as a KL control problem for a non-Markovian system.
- They treat a trained MLE sequence model as the prior policy, and thus the objective is to train a new policy to maximize some rewards while keeping close to the original MLE model.
  - $\tau = \{a_1, a_2, \ldots, a_{t-1}\}$: the sequence, $\gamma(\tau)$: the reward of the sequence, $p(\tau)$: the prior distribution over τ given by the trained sequence model, $q(\tau)$: the policy of the Sequence Tutor model:

    $$L(q) = E_{q(\tau)}[\gamma(\tau)/c - D_{KL}[q(\tau)||p(\tau)].$$

  - The reinforcement learning objective

    $$L(\theta) = E_{\pi}[\sum_t r(s_t, a_t)/c + \log p(a_t|s_t) - \log_{\pi_\theta}(a_t|s_t)]$$

    $E_{\pi}[\cdot]$:expectation with respect to sequences sampled from π.
  - Derive two algorithm to parameterize π_θ.

Experiments

- Generation of Melody and Molecular
- Compare three methods for implement the Sequence tutor:
  - Q-learning with the deterministic policy.
  - two methods for KL-control with the non-deterministic policy.
- Compare the RL-only with no prior policy and MLE RNN.

Conclusions

- Similar methods can be applied on text generation if a deterministic policy is applied.