Neural Machine Translation using Sequence Level Training

Kyu-Young Kim
Department of Computer Science, Stanford University

Talk at https://youtu.be/HfPee5AWFhw

Problem

RNN models are typically trained with per-token, cross-entropy loss using the ground truth sequence often referred to as the maximum likelihood estimation (MLE) or teacher-forcing. This poses two problems.

- [**Exposure bias**] The distribution the model is conditioned on during training is different than that during inference.
- [**Loss mismatch**] The loss function that the model was trained to optimize for is different than the metric used to evaluate the model.

Related Works

- [**Beam search**] Maintain a set of candidate sequences during the decoding stage and select the one with the highest score at the end of generation. Finds a higher quality sequence but is significantly slower.
- [**Scheduled sampling**] (Bengio, et al., 2015) Curriculum learning approach where at each RNN timestep flip a coin to decide whether to use the ground truth token or the model's own prediction as an input for the subsequent timestep.

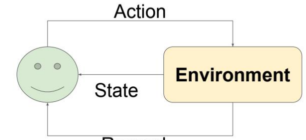
Sequence-to-Sequence Model

- [**Encoder**] Maps an input sequence into a fixed-sized vector representation.
- [**Decoder**] Takes the encoder output as generates a sequence one token at a time.
- [**Training**] Provides the ground truth sequence as input to the decoder. Typically uses cross-entropy loss:

$$loss = -\frac{1}{m}\sum_{t=1}^{T'}\sum_{c=1}^{s} y_{t(c)} \log(l_{t(c)})$$

- [**Inference**] Uses the model's own prediction from the previous step as an input.

RL of Sequence Learning



- View the RNN decoder as an agent and the hidden state as the environment.
- The output token generated by the decoder is the action the agent takes and it receives a reward as computed by an evaluation metric.

$$\nabla loss = -\sum_{i=1}^{m} \mathbb{E}_{\pi(Y'^i|X^i)}[R(Y'^i|Y^i)\nabla \log \pi(Y'^i|X^i)]$$

Experiments and Analysis

- [**Dataset**] German-to-English text translation from TED and TEDx talks.
- [**Vanilla seq2seq**] Used basic LSTM cell with varying number of hidden units. More model capacity and regularization important.
- [**Attention model**] Used layer-normalized LSTM cell with dropout applied to input and output. Added a decoder attention to encoder states. Better generalization and model convergence.
- [**RL model**] Curriculum learning to gradually expose the model to its own predictions and incorporate BLEU score into the loss.
- [**Future work**] Curriculum learning schedule and model convergence. Length of the sequences to learn and the effectiveness of the RL method.