Sequence-Level Training

April 5, 2017

Jindřich Helcl, Jindřich Libovický
Word-Level Training

- Likelihood of a sentence in word-level training:

\[ p(y|x, \theta) = \prod_{t=1}^{T} p(y_t|y_{<t}, x, \theta) \]

- Log-likelihood as a loss function:

\[ \mathcal{L}(\theta) = \sum_{i=1}^{N} \log p(y_i|x_i, \theta) \]

\[ = \sum_{i=1}^{N} \sum_{t=1}^{T_i} \log p(y_{it}|y_{i,<t}, x_i, \theta) \]
Log-likelihood

\[
\mathcal{L}(\theta) = \sum_{i=1}^{N} \sum_{t=1}^{T_i} \log p(y_{it} | y_{i,<t}, x_i, \theta)
\]

- Fast, yields good results
- Differentiable!

\[
\frac{\partial \mathcal{L}(\theta)}{\partial \theta_i} = \sum_{i=1}^{N} \sum_{t=1}^{T_i} \frac{\partial p(y_{it} | y_{i,<t}, x_i, \theta)}{p(y_{it} | y_{i,<t}, x_i, \theta)}
\]
Log-likelihood – problems

- Training objective different from the evaluation metric
- Loss function defined on word-level
- Output word distribution compared with an one-hot distribution
- Suffers from exposure bias
Sentence-level Losses

- Score the output sentence as a whole
- Solve the exposure bias problem
- Many metrics out there: BLEU, METEOR, ...
- Good correlation with human judgement

- Drawback: Although differentiable, the derivatives are locally constant

- Solution?
Sentence-level Training

- Problem: locally constant derivatives
- Cause: selecting the best word during decoding
- Idea: Score the distribution over possible sentences rather than the model output
- Can we still somehow use metrics like BLEU?
Sentence-level Training

Let the loss be the expected value of the scoring function $r$ over all possible outputs $\mathcal{Y}$ w. r. t. reference translation $y^*$:

$$\mathbb{E}_{y \in \mathcal{Y}}[r(y, y^*)] = \sum_{y \in \mathcal{Y}} p(y|x, \theta)r(y, y^*)$$

good model gives high score to good sentences, low score to bad sentences
Minimum Risk Training (1)

Shen et al., 2016 (https://arxiv.org/abs/1512.02433)

▶ Minimum risk training uses the expected score to calculate the risk:

\[ \mathcal{R}(\theta) = \sum_{i=1}^{N} \mathbb{E}_{y \in Y} [r(y, y^*)] \]

▶ Nice derivative
Problem: The space of all possible outputs $\mathcal{Y}(x)$ for input sentence $x$ is way too large

Approximate the expected value by using only a few samples from the distribution

$$|\mathcal{Y}(x)| \gg |S(x)| = \text{usually around 100}$$
Resampling

\[
\tilde{R}(\theta) = \sum_{s=1}^{S} \mathbb{E} \Delta(y, y^*)
\]

\[
\approx \sum_{s=1}^{S} \sum_{y \in \mathcal{S}(x)} Q(y|x, \theta, \alpha) \Delta(y, y^*)
\]

\[
Q(y|s, \theta, \alpha) = \frac{p(y|s, \theta)^{\alpha}}{\sum_{y' \in \mathcal{S}(x)} p(y'|x, \theta)}
\]
Sequence-level Training using Reinforcement Learning

Ranzato et al., 2015 (https://arxiv.org/abs/1511.06732)

- In minimum risk training, we approximate the expected score
- Here: approximate the gradients

\[
\frac{\partial L}{\partial \theta} = \sum_t \frac{\partial L}{\partial o_t} \cdot \frac{\partial o_t}{\partial \theta}
\]

- Use the REINFORCE algorithm (Williams, 1992):

\[
\frac{\partial L}{\partial o_t} \approx (r(y, y^*) - \bar{r}_{t+1}) (p(y_{t+1}|y_{<t}, x, \theta) - \mathbb{1}(y_{t+1}))
\]