NPFL116 Compendium of Neural Machine Translation

Sequence-Level Training

April 5, 2017

Jindřich Helcl, Jindřich Libovický

Charles Univeristy in Prague
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

Word-Level Training

- Likelihood of a sentence in word-level training:

$$\mathrm{p}(\mathbf{y}|\mathbf{x},\theta) = \prod_{t=1}^{T} \mathrm{p}(y_t|\mathbf{y}_{<t},\mathbf{x},\theta)$$

- Log-likelihood as a loss function:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= \sum_{i=1}^{N} \log \mathrm{p}(y_i|x_i,\theta) \\ &= \sum_{i=1}^{N} \sum_{t=1}^{T_i} \log \mathrm{p}(y_{it}|\mathbf{y}_{i,<t},\mathbf{x}_i,\theta)\end{aligned}$$

Log-likelihood

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \sum_{t=1}^{T_i} \log \mathrm{p}(y_{it} | \mathbf{y}_{i,<t}, \mathbf{x}_i, \boldsymbol{\theta})$$

- Fast, yields good results
- *Differentiable!*

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_i} = \sum_{i=1}^{N} \sum_{t=1}^{T_i} \frac{\partial\, \mathrm{p}(y_{it} | \mathbf{y}_{i,<t}, \mathbf{x}_i, \boldsymbol{\theta}) / \partial \boldsymbol{\theta}_i}{\mathrm{p}(y_{it} | \mathbf{y}_{i,<t}, \mathbf{x}_i, \boldsymbol{\theta})}$$

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 $y^\star$:

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

- 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:

$$\mathcal{R}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \mathop{\mathbb{E}}_{y \in \mathcal{Y}} \left[r(y, y^{\star}) \right]$$

- Nice derivative

Minimum Risk Training (2)

- 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

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

Resampling

$$\bar{R}(\theta) = \sum_{s=1}^{S} \mathbb{E}\, \Delta(y, y^\star)$$

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

$$Q(y|s, \theta, \alpha) = \frac{\mathrm{p}(y|s, \theta)^\alpha}{\sum_{y' \in \mathcal{S}(x)} \mathrm{p}(y'|x, \theta)}$$

Sequence-level Training using Reinforcement Learning

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

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

$$\text{chain rule:} \quad \frac{\partial \mathcal{L}}{\partial \theta} = \sum_t \frac{\partial L}{\partial \mathbf{o}_t} \cdot \frac{\partial \mathbf{o}_t}{\partial \theta}$$

- Use the REINFORCE algorithm (Williams, 1992):

$$\frac{\partial \mathcal{L}}{\partial \mathbf{o_t}} \approx \left(r(y, y^\star) - \bar{r}_{t+1}\right)\left(\mathrm{p}(y_{t+1}|\mathbf{y}_{<t}, \mathbf{x}, \boldsymbol{\theta}) - \mathbb{1}(y_{t+1})\right)$$