Homework 3: Recurrent Neural Networks and Attention

Deep Learning (84100342-0)
Spring 2020
Tsinghua University

1 Introduction

As we have learned in class, RNNs can be applied to many sequence models across machine translation to image captioning. Language modeling is a central task in NLP and language models can be found at the heart of speech recognition, machine translation, and many other systems. In this homework, you are required to solve a language modeling problem by designing and implementing recurrent neural networks (RNNs).

A language model is a probability distribution over sequences of words. Given such a sequence $(\mathbf{x}_1, ..., \mathbf{x}_m)$ with length m, it assigns a probability $P(\mathbf{x}_1, ..., \mathbf{x}_m)$ to the whole sequence. In detail, given a vocabulary dictionary of words $(\mathbf{v}_1, ..., \mathbf{v}_m)$ and a sequence of words $(\mathbf{x}_1, ..., \mathbf{x}_t)$, a language model predicts the following word $\mathbf{x}_{t+1}$ by modeling: $P(\mathbf{x}_{t+1} = \mathbf{v}_j | \mathbf{x}_1, ..., \mathbf{x}_t)$ where $\mathbf{v}_j$ is a word in the vocabulary dictionary. Conventionally, we evaluate our language model in terms of perplexity (PP) `https://en.wikipedia.org/wiki/Perplexity`. Note that, $\mathrm{PP} = \exp(\text{Average Cross Entropy Loss})$.

2 Dataset

- The dataset have two parts: train set and valid set.
- Directory structure:
  - "./src/" contains the start code
  - "./data/" contains the train set and the valid set.

3 Requirements and Evaluations

3.1 Programming Language

Python only.

3.2 Deep Learning Framework

We recommend PyTorch and TensorFlow. If using other frameworks, please contact TA.

3.3 Tutorials

- **Lab3 of this course**
  - RNN From Scratch
  - Sentiment-RNN
  - Advanced RNNs

3.4 Scoring

- Construct your own RNNs, train your model from scratch (fine-tuning is forbidden) using the recommended deep learning framework. (**40%**)
- Validate your model on the valid set, and report training and validation curves. (**40**%)
- Use extra techniques you find in other materials to further improve your model. Please explain why you choose it and how it works. (**10**%)
- Develop a kind of attention mechanism (temporal attention or self attention) and see whether it is able to improve your model (**10**%). If the attention mechanism you choose doesn't work well, please explain why.

3.5 BONUS

- For constructing RNN, you are allowed to use well-established libraries, such as torch.nn. However, it is a **BONUS** (**10**%) if you implement the RNN network with basic arithmetic operators (e.g. torch., torch.mm, torch.cat). Please highlight it in your report if you have finished this task as a bonus.
- To be honest, both the provided *pre-processing* and the provided *evaluation* have some flaws. If you are able to write your own code with some better ideas for *pre-processing and evaluation*, you can get a bonus (**10**%) with regard to what you implement. For example, **BLEU**, **METEOR**, **ROUGE** are also good evaluation methods. Still, please highlight it in your report if you have finished this task as a bonus.

3.6 Notification

- We have provided the start code to concentrate your attention on the construction and training of RNN itself.
- Please submit your *code and report* as an Archive (zip or tar). The document is supposed to cover your **insights** of the proposed model, the **technical details**, the **experimental results** (including training and validation curves), and the necessary **references**. You are forbiddened to use additional data sources.
- We will focus on your code and document to decide your score. Still, under equal conditions (novelty, code quality, document quality), a higher performance along with reasonable computation efficiency contributes a higher score.