Homework 3: Recurrent Neural Networks and Transformer

Deep Learning (84100343-0)
Spring 2021
Tsinghua University

1 Introduction

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), as well as Transformer. 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

- We adopt WikiText2 as the dataset, which consists of three parts:
  - train set: "`data_loader.train_data`"
  - validation set: "`data_loader.val_data`"
  - test set: "`data_loader.test_data`"

3 Requirements

3.1 Programming Language

Python only.

3.2 Deep Learning Framework

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

4 Tutorials

- **Lab3 of This Course**
  - RNN From Scratch
  - Sentiment-RNN
  - Advanced RNNs

5 Tasks and Scoring

You need to finish the following tasks on the given dataset:

- **Task A**: Construct a kind of standard RNN (*e.g.* LSTM, GRU, etc.) and train your model from scratch using the recommended deep learning framework. Note that fine-tuning from a large-scale pre-trained model is forbidden. Validate your model on the *valid* set, and report training and validation curves. Evaluate the best model on the *test* set. [**30pts**]
- **Task B**: Construct the standard Transformer (Attention is All You Need) and train your model from scratch using the recommended deep learning framework. Still, fine-tuning from a large-scale pre-trained model is forbidden. Validate your model on the *valid* set, and report training and validation curves. Evaluate the best model on the *test* set. [**30pts**]

- **Data Preparation**:
  - We have provided the data preparation code "`data.py`". By running "`python main.py`", you will see how the data is preprocessed. Please summarize the differences between preprocessing text data for language modeling with preprocessing image data for object recognition. [**10pts**]
- **Technical Details**:
  - As we have learned in class, mask is used in Transformer to keep it autoregressive. Please explain why mask is necessary in Transformer and how it is implemented in your code. [**10pts**]
- **Attention Visualization**:
  - Please visualize the attention values of some typical words and check whether the Transformer can learn meaningful attention values. [**10pts**]
- **Extra Techniques**:
  - Please adopt an extra technique you find in other materials to further improve your model. Please explain why you choose it, check whether it works on our given dataset and give a convincing reason. [**10pts**]

6 Notifications

- 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**.
- We will focus on your code and document to decide your score. Still, under equal conditions including novelty, code quality, and document quality, a better performance along with reasonable computation efficiency contributes a higher score.