

Deep Learning Homework 4

Sequence modeling

due on May 14, 2021

In this homework, you need to implement LSTM and Transformer-based models for generation and classification tasks. Please clone the starter code from https://github.com/xssstory/dl_course_hw4. Please submit all the code, the best models and your report to web learning.

1 Generation

In this task, you need to train sequence-to-sequence models for conditional generation and language models for unconditional generation.

1.1 Writing Couplets with Sequence-to-Sequence Models

Sequence-to-sequence aims to train a model

$$P(Y|X) = \prod_{t=1}^L P(Y_t | f_{dec}(Y_{i < t}), f_{enc}(X))$$

on parallel data (X, Y) , where f_{enc} and f_{dec} are encoder and decoder respectively. In this task, we provide a couplet corpus, and your need to:

1. Complete the code of `Seq2SeqModel` in `generation/lstm.py`. You also need to enhance it with attention mechanism and show the ablation study of attention in your report.
2. Complete the code for modules in `generation/transformer.py` and train a sequence-to-sequence model with transformer architecture.
3. Complete the code in function `generate` of `Seq2SeqModel` in `generation/lstm.py` and `generation/transformer.py` with beam search.
4. Report the training curves, ablation studies and generated samples.

1.2 Writing Poems with Language Models

Language models learn the sequence distribution in an autoregressive manner:

$$P(X; \theta) = \prod_{t=1}^L P(X_t | X_{i < t}; \theta).$$

In this task, you are given a Chinese classical poetry corpus. You need to

1. Complete the code of `LModel` in `generation/lstm.py` and `generation/transformer.py` (construct the model and implement function `generate` with beam search).
2. Report training curves, generated samples and (optional) ablation studies.

1.3 Tips:

1. Make sure you can run `generation/evaluation.py`. We expect 4 models named “`lstm_lm.pt`”, “`lstm_seq2seq.pt`”, “`transformer_lm.pt`” and “`transformer_seq2seq.pt`” in this problem.
2. You can modify `generation/dataset.py` for training if necessary, but make sure we can use the original `dataset.py` for inference.
3. You are encouraged to use pre-trained word embedding and pre-trained models to improve the performance, but keep in mind that we will **not** install extra packages when testing your model.
4. We will grade this task according to the quality of your generated samples and the perplexity in the test set.

2 Classification

In this problem, we provide a reading comprehension dataset. There are thousands of articles in the dataset, and each article has several questions. For each question, there are 2 to 4 choices and only one choice is correct. You need to build and train a model to choose the correct answer.

Requirements The only requirement is to make sure you can run `classification/evaluation.py`. We will grade your model according to the accuracy in the test set. You will get all points if your test accuracy > 0.6 .

Bonus The best submission with the highest testing accuracy will get 2 bonus points for the final course grade.

Human evaluation We also invite you for providing human answers to this classification challenge. Participants with **honest** feedback will get one extra point. A questionnaire for human evaluation will be released soon.