Week 2 Assignment [4 Points]

Generative AI

Saarland University – Winter Semester 2024/25

Submission due: 28 October 2024 by 6pm CET

genai-w24-tutors
genai-w24-tutors@mpi-sws.org

1 Reading Assignment

This week's reading assignment comprises of the following material:

(R.1) Chapters 7.0 (i.e., introduction), 7.1, 7.3 (without 7.3.1), 7.6, and 7.7 from the book by Jurafsky and Martin [1]. The book is available freely online at the following link: https://web.stanford.edu/~jurafsky/slp3/ed3bookaug20_2024.pdf.

(R.2) Chapters 9.0 (i.e., introduction), and 9.1 (without the part on multi-head attention) from the book by Jurafsky and Martin [1].

[OPTIONAL] Below, we are providing additional reading material covering the research papers relevant to the lecture:

- A Neural Probabilistic Language Model [2, 3]. Paper link: https://papers.nips.cc/paper_files/paper/2000/file/728f206c2a01bf572b5940d7d9a8fa4c-Paper.pdf.
- Sequence to Sequence Learning with Neural Networks [4]. Paper link: https://proceedings.neurips.cc/paper_files/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.
- Neural Machine Translation by Jointly Learning to Align and Translate [5]. Paper link: https://arxiv.org/pdf/1409.0473.
- Show, Attend and Tell: Neural Image Caption Generation with Visual Attention [6]. Paper link: http://proceedings.mlr.press/v37/xuc15.pdf.
- Attention is All You Need [7]. Paper link: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

2 Exercise Assignment

This week's exercise assignment aims to give you insights into the feedforward neural language model architecture and the attention head. These questions should be answered in the PDF submission file – see instructions in Section 4.

(E.1) Consider the neural architecture in the Week 2's lecture slide #27, titled *Simple Feedforward Neural LM: Detailed Architecture*. For this question, we consider a general setting where the input to the model is $N-1$ words denoted by $\mathbf{x}_{t-N+1}, \ldots, \mathbf{x}_{t-2}, \mathbf{x}_{t-1}$ and represented as one-hot encoded vectors. The vocabulary size is $|V|$, the embedding dimension is d, and the number of neural units in the hidden layer is d_h. Write down detailed steps to find the total number of parameters in this network architecture.

HINT 1: The total parameters include parameters for the embedding matrix E, parameters for d_h neural units, and parameters for the unembedding matrix U.

HINT 2: Calculate the dimensions of the intermediate vectors $\mathbf{e}$, $\mathbf{h}$, and $\mathbf{z}$.

(E.2) Consider the single attention head in Week 2's lecture slide #38, titled *Attention Mechanism with Single Head: Actual Version*. The input to the attention head is $N-1$ embedding vectors denoted by $\mathbf{e}_{t-N+1}, \ldots, \mathbf{e}_{t-2}, \mathbf{e}_{t-1}$. The attention mechanism uses parameters corresponding to query matrix W^q (size $d_k \times d$), key matrix W^k (size $d_k \times d$), and value matrix W^v (size $d_v \times d$). Note that d, d_k, and d_v could be different from each other. Write down detailed steps to compute the attention vector $\mathbf{a}$ as output from the attention head.

(E.3) Similar to the above question, again consider the single attention head in Week 2's lecture slide #38, *Attention Mechanism with Single Head: Actual Version*. Note that d, d_k, and d_v could be different from each other. Write down detailed steps to find the total number of parameters in this attention head.

(E.4) Consider the neural architecture in the Week 2's lecture slide #39, titled *Incorporating Attention Mechanism in Simple Ff Neural LM*. The input to the architecture is $N-1$ words denoted by $\mathbf{x}_{t-N+1}, \ldots, \mathbf{x}_{t-2}, \mathbf{x}_{t-1}$ and represented as one-hot encoded vectors. The vocabulary size is $|V|$, the embedding dimension is d, and the number of neural units in the hidden layer is d_h. Moreover, the attention head uses query matrix W^q (size $d_k \times d$), key matrix W^k (size $d_k \times d$), and value matrix W^v (size $d_v \times d$). Note that d, d_k, and d_v could be different from each other. Write down detailed steps to find the total number of parameters in this network architecture.

HINT 1: The total parameters include parameters for the embedding matrix E, parameters for the attention head, parameters for d_h neural units, and parameters for the unembedding matrix U.

HINT 2: Calculate the dimensions of the intermediate vectors $\mathbf{e}$, $\mathbf{a}$, $\mathbf{h}$, and $\mathbf{z}$.

3 Implementation Assignment

For this week's implementation assignment, you will run pretrained transformer models and evaluate them. The transformer models are already trained on the `processed_berkeley_restaurant.txt` corpus from Week 1 and provided as part of the assignment. We have provided the following files in `week2_implementation.zip`:

- `lm_transformer.py`: An implementation of a transformer language model. It can load a pretrained transformer model and evaluate the loaded model.
- `processed_berkeley_restaurant.txt`: This is the corpus from Week 1 assignment and the provided transformer models are already trained on this corpus.
- `eval_berkeley_restaurant.txt`: This file contains a set of sentences that are used for evaluating the provided models using perplexity metric.
- `transformer_model_N{N}_layers{LAYERS}.pth`: A pretrained transformer model that is trained on the `processed_berkeley_restaurant.txt` corpus with `N`$= \{3, 5\}$ and `LAYERS`$= \{2, 4\}$. Here, `N` is a hyperparameter which means that the number of input words (size of context) is `N-1`; layers is a hyperparameter that corresponds to the number of stacked transform blocks.

As part of this assignment, you will simply execute the provided code by modifying the inputs and don't have to implement any new code. More concretely, in the `__main__` function of `lm_transformer.py`, you will vary the value of `N` and `LAYERS`.[1] When the code is executed, it first loads the transformer model `transformer_model_N{N}_layers{LAYERS}.pth`, prints the total number of parameters, computes and prints the perplexity on `eval_berkeley_restaurant.txt`, and then generates and prints ten sentences using this model. After running the code for the four provided models with different `N` and `LAYERS`, answer the following questions in the PDF submission file – see instructions in Section 4.

(I.1) Report the perplexity for these four model configurations as you vary `N` and `LAYERS`. Briefly discuss how changes in these hyperparameters affect the perplexity results.

(I.2) Report the total number of parameters for these four model configurations as you vary `N` and `LAYERS`. Briefly discuss how changes in these hyperparameters affect the total number of model parameters.

[1] In case you are using Colab, you will have to first upload all the provided files and models to Colab.

4 Submission Instructions

Please submit the following files using your personal upload link:

- **<Lastname>_<Matriculation>_week2.pdf**. This PDF file will report answers to the following questions: E.1, E.2, E.3, E.4, I.1, and I.2. The PDF file should be generated in LaTeX using NeurIPS 2024 style files; see further formatting instructions in Section 5.1. The PDF file size should **not exceed** 1**mb**. Please use the following naming convention: Replace <Lastname> and <Matriculation> with your respective Lastname and Matriculation number (e.g., Singla_1234567_week2.pdf).

5 Technical Instructions

5.1 Instructions for Preparing PDFs

Refer to Week 1 assignment.

5.2 Instructions for Implementation

Refer to Week 1 assignment.

References

[1] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd edition, 2024. Online manuscript released August 20, 2024; https://web.stanford.edu/~jurafsky/slp3/.

[2] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A Neural Probabilistic Language Model. In *NeurIPS*, 2000.

[3] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A Neural Probabilistic Language Model. *JMLR*, 2003.

[4] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to Sequence Learning with Neural Networks. In *NeurIPS*, 2014.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR*, 2015.

[6] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *ICML*, 2015.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All You Need. In *NeurIPS*, 2017.

Based on template from https://media.neurips.cc/Conferences/NeurIPS2024/Styles.zip