Prof. Dr. Goran Glavaš,
M.Sc. Fabian David Schmidt
M.Sc. Benedikt Ebing
Lecture Chair XII for Natural Language Processing, Universität Würzburg

1. Exercise for "Multilingual Natural Language Processing"

14.06.2024

1 Paper Readings

The PEFT literature is vast and grows rapidly. The papers listed below serve as an initial starting point for your reading to complete the homework.

- Towards A Unified View of Parameter-Efficient Transfer Learning
- MAD-X: An Adapter-Based Framework For Multi-Task Cross-Lingual Transfer
- LoRA: Low-Rank Adaption of Large Language Models
- Prefix-Tuning: Optimizing Continuous Prompts for Generation

2 Parameter-Efficient Fine-Tuning: Basics

1. Describe the core idea of parameter-efficient fine-tuning (PEFT) briefly.
2. Concisely explain the key advantages of PEFT!
3. Can you think of and explain potential disadvantages oft PEFT?

3 Comparison of methods

Analyse and compare (i) LoRA, (ii) Prefix-Tuning, and (iii) Adapters along the following dimensions:

- Modelling: how are the original language model representations updated during PEFT between the approaches?
- Implementation, ease of use
- Inference

4 Additional Exercises (not required for bonus)

1. Given the following two cross lingual word embedding spaces W_{de} (the rows represent the embeddings for the words "Hallo", "Welt", "Programm") and W_{en} (the rows represent the embeddings for the words "hello", "world", "program"), as well as the projection matrix $W_{de \to en}$, project the German words into the English embedding space and compute the "least squares" loss.

$$W_{de} = \begin{bmatrix} -0.7085 & -0.8577 & 1.3898 \\ -0.4311 & -1.6566 & -1.9603 \\ 1.5719 & -0.5320 & -0.7148 \end{bmatrix} \quad W_{en} = \begin{bmatrix} 0.5787 & -1.4529 & -0.1379 \\ 0.3200 & 0.4139 & 1.7434 \\ 0.4766 & 1.5077 & -0.0707 \end{bmatrix}$$

$$W_{de \to en} = \begin{bmatrix} -0.8262 & -0.2042 & -1.8096 \\ -0.7249 & -1.2263 & -0.5002 \\ -0.8531 & -1.3717 & 0.9136 \end{bmatrix}$$

Solution:

Result of the projection $W_{de} W_{de \to en}$:

$$W_{de \to en} = \begin{bmatrix} 0.0216 & -0.7099 & 2.9810 \\ 3.2295 & 4.8086 & -0.1822 \\ -0.3033 & 1.3120 & -3.2315 \end{bmatrix}$$

Loss: 12.1270

2. Compute the output representations of the i-th layer of an (a) encoder and (b) decoder only Transformer (without Layer Normalization). The input from the (i-1)-th layer is given as follows:

$$
\begin{aligned}
hello &\rightarrow [0.8109, -0.9391, 0.2519] \\
world &\rightarrow [2.1968, 0.4785, 1.5207] \\
! &\rightarrow [-0.3264, 0.1585, 0.8469]
\end{aligned}
$$

The projection matrices for queries W_q, keys W_k and values W_v are as follows:

$$
W_q = \begin{bmatrix} 0.8571 & 0.1942 & -1.1185 \\ 1.6184 & -0.2665 & -0.9808 \\ -1.2820 & -0.1892 & 0.7269 \end{bmatrix}; W_k = \begin{bmatrix} 0.2176 & -0.2462 & 0.0557 \\ 0.6926 & -2.0236 & -0.2968 \\ -0.6658 & -2.5785 & -0.4534 \end{bmatrix};
$$

$$
W_v = \begin{bmatrix} -0.2016 & 1.4234 & -0.1733 \\ 1.3098 & -0.4815 & 0.0123 \\ 0.1350 & 0.3998 & 0.4573 \end{bmatrix}
$$

The matrices for the feed-forward layer are as follows:

$$
W_1 = \begin{bmatrix} 0.5325 & 0.7081 & -1.1454 \\ 0.5511 & 0.6274 & -0.8078 \\ 1.0224 & 0.4148 & -0.4800 \end{bmatrix}; W_2 = \begin{bmatrix} -1.3081 & 1.9878 & 2.6332 \\ 2.6049 & 0.9369 & -0.9614 \\ 0.2293 & 0.3262 & 0.7968 \end{bmatrix};
$$

For simplicity, we omit the bias terms and use a single attention head. The activation function is Relu.

Solution Encoder Only:

$$
\text{(output of self-attention layer) } Z = \begin{bmatrix} -0.7098 & 1.4047 & 0.1288 \\ 0.1002 & 1.9549 & 0.3040 \\ -0.1147 & 1.7538 & 0.2594 \end{bmatrix}
$$

$$
output = \begin{bmatrix} 0.5448 & 2.3451 & 1.7216 \\ 6.6879 & 14.9026 & 9.7301 \\ 0.5152 & 7.0498 & 4.9465 \end{bmatrix}
$$

Solution Decoder Only:

$$\text{(output of self-attention layer) } Z = \begin{bmatrix} -1.3596 & 1.7072 & -0.0369 \\ -0.0467 & 3.0566 & 0.2315 \\ -0.1147 & 1.7538 & 0.2594 \end{bmatrix}$$

$$output = \begin{bmatrix} -0.5321 & 1.6367 & 0.9634 \\ 7.3975 & 17.4307 & 10.3197 \\ 0.5152 & 7.0499 & 4.9466 \end{bmatrix}$$

3. How do the computations of the previous task change, if you apply Prefix-Tuning, a bottleneck Adapter after the feed-forward layer, or Lora on the projection matrices (W_q, W_k, W_q)