DNLP-assignment2

Name: , Student ID:

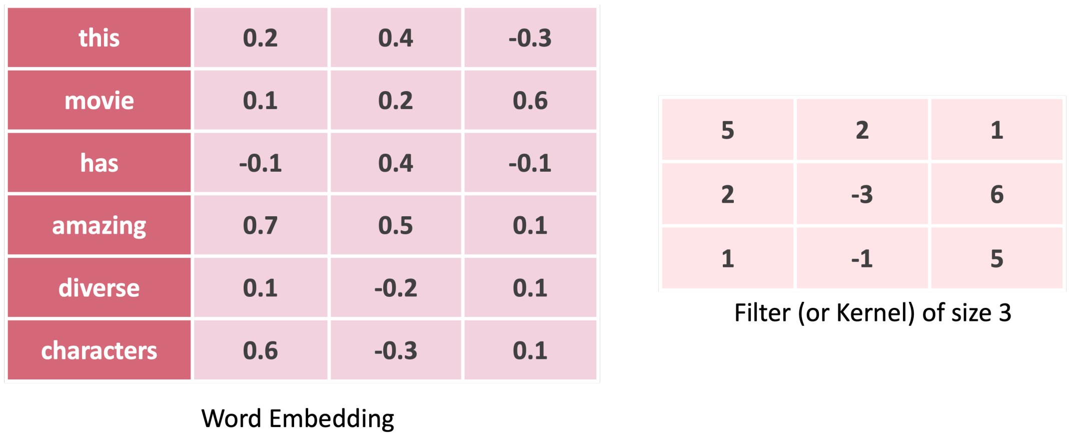
April 2023

Introduction

This assignment consists of two parts. In part 1, you will be asked to solve exercise problems. In part 2, you will implement the attention mechanism.
Please submit your report of part 1 as a pdf and name the file "assignment2_written".
In part 2, submit your implementations as zip file "assignment2_code".
Please bundle the "assignment2_written.pdf" and "assignment2_code.zip", and submit as one zip file with name 'your student number_your name'.

1. Part 1 (40 Points)

1.1 Problem 1



this	0.2	0.4	-0.3
movie	0.1	0.2	0.6
has	-0.1	0.4	-0.1
amazing	0.7	0.5	0.1
diverse	0.1	-0.2	0.1
characters	0.6	-0.3	0.1

Word Embedding

5	2	1
2	-3	6
1	-1	5

Filter (or Kernel) of size 3

Figure 1: Problem 1 figure

1. We learned about 1D convolution for text in class. Draw a table of results for each condition by referring to Fig 1. [6 points]

   (i) Apply the filter with stride 1 without padding. [2 points]

(ii) Apply the filter with stride 2 with padding size 1 at the beginning and end. [2 points]

(iii) Apply the 2-max pooling to (i) and (ii), respectively. [2 points]

2. Explain why sub-word modeling is useful in natural language processing. [2 points]

3. The attention flow layer is central idea of improvements to the BiDAF architecture. [6 points]

   (i) What is the attention flow idea? [2 points]

   (ii) What is the Context-to-Question attention? [2 points]

   (iii) What is the Questio-to-Context attention? [2 points]

4. Why multi-headed self-attention can be preferable to single-headed self-attention? [2 points]

5. In the class, we learned beam search decoding. [4 points]

   (i) If the beam size k is small, what happens? Is this problem? [2 points]

   (ii) If the beam size k is large, what happens? Is this problem? [2 points]

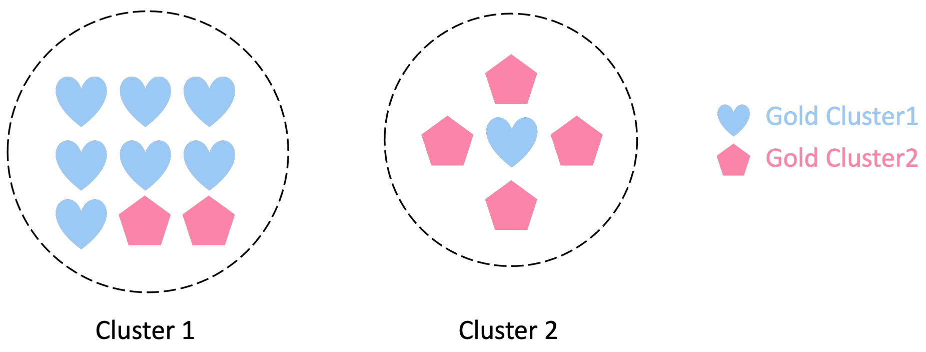


Figure 2: Cluster

6. There are some evaluation metrics in the natrual language generation. [6 points]

   (i) Write the definition of ROUGE. [2 points]

   (ii) Write the definition of BLEU.[2 points]

   (iii) Explain the difference between ROUGE and BLEU [2 points].

7. Coreference Resolution[4 points]

   (i) What is coreference resolution? [2 points]

   (ii) Compute the B-cubed in Fig 2. [2 points]

8. In Transformers, we perform *self-attention*, which roughly means that we draw the keys, values, and queries from the same data. More precisely, let $\{x_1, ..., x_n\}$ be a sequence of vectors in $\mathbf{R}^d$. Think of each x_i as representing word i in a sentence. One form of self-attention defines keys, queries, and values as follows. Let $V, K, Q \in \mathbf{R}^d \times d$ be parameter matrices. Then

$$v_i = Vx_i \quad i \in \{1, ..., n\}$$
$$k_i = Kx_i \quad i \in \{1, ..., n\}$$
$$q_i = Qx_i \quad i \in \{1, ..., n\}$$

Then we get a context vector for each input i, we have $c_i = \sum_{j=1}^{n} \alpha_{ij} v_j$, where α_{ij} is defined as $\alpha_{ij} = \frac{\exp(k_j^\top q_i)}{\sum_{l=1}^{n} \exp(k^t op_l q_i)}$. Note that this is single-based self-attention.
In this questions, we will show how key-value-query attention like this allows the network to use different aspects of the input vectors x_i in how it defines keys, queries, and values. Intuitively, this allows networks to choose different aspects of x_i to be the "content" (value vector) versus what it uses to determine "where to look" for content (keys and queries). [10 points]

(i) First, consider if we did not have key-query-value attention. For keys, queries, and values we'll just use x_i; that is, $v_i = q_i = k_i = x_i$. We will consider a specific set of x_i. In particular, let u_a, u_b, u_c, u_d be mutually orthogonal vectors in $\mathbf{R}^d$, each with equal norm $\|u_a\| = \|u_b\| = \|u_c\| = \|u_d\| = \beta$, where β is very large. Now, we let our x_i be :

$$x_1 = u_d + u_b$$
$$x_2 = u_a$$
$$x_3 = u_c + u_b$$

If we perform self-attention with these vectors, what vector does c_2 approximate? Would it be possible for c_2 to approximate u_b by adding either u_d or u_c to x_2? Explain why or why not (either math or English is fine). [5 points]

(ii) Now consider using key-query-value attention as we've defined it originally. Using the same definitions of x_1, x_2 and x_3 in part (i), specify matrices K, Q, V such that $c_2 \approx u_b$, and $c_1 \approx u_b - u_c$. There are many solutions to this problem, so it will be easier for you, if you first find V such that $v_1 = u_b$ and $v_3 = u_b - u_c$, then work on Q and K. Some outer product properties may be helpful (as summarized in this footnote) [1] [5 points].

2. Part 2 (10 Points)

We learned the attention mechanism in the class. In this part, you implement the `attention.py`. (Note: You may install the `pytorch`.)

1. Implement `score` in `attention.py`. It recevies (batched) query vectors $Q \in \mathbf{R}^{B \times T' \times T}$ where

$$L[l]_{i,j} = Q[l]_{i,:}^\top W K[l]_{j,:} \qquad \forall l = 1, ..., B$$

where $W \in \mathbf{R}^{d \times d}$ is a parameter of the model. Multiplying by W corresponds to feeding Q to the linear layer already in the model `self.linear_in`, so do that. You should not use any for loops. You can do this using **torch.bmm** (batched matrix multiplication, look it up). You will want to change/switch dimensions of the input tensors efficiently by using `view` and `transpose`.

2. Implement Eq.(5) of Luong et al. (2015) [1] in `forward` in `attention.py`. The c_t is already given (assuming `score`). The h_t is just `queries`. The multiplication by W_s should be done by feeding (c_t, h_t) to `self.linear_out` already in the model. You will want to use `torch.cat, view, torch.tanh` to align dimensions correctly.

3. Pass the simple tests in `test.attention.py` to check if you did the previous two problems correctly. Please capture the completed screen and submit it with the assignment.

[1]For orthogonal vecotrs $u, v, w \in \mathbf{R}^d$, the outer product $uv^\top$ is a matrix in $\mathbf{R}^{d \times d}$, and $(uv^\top)v = u(v^\top v) - u\|v\|_2^2$, and $uv^\top)w = u(v^\top w) = u * 0(\because v \perp w)$

References

[1] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.