Deep Learning in NLP

Homework 1

Solution to be sent to `waszczuk@phil.hhu.de` and `cwurm@phil.hhu.de` by 9.11.2020 (included, i.e. can be sent till Monday in the evening). It is allowed to do the homework in groups (max. 3 persons per group). Please specify in the email (or in an accompanying README) the authors of the solution.

The archive file with one Python file per exercise (`ex1.py`, `ex2.py`, ...) can be found on the course's website. The missing pieces in the code are marked with `TODO`. Please do not modify the docstring tests, they will be used for evaluation. The solution sent by email should have the same form: it should be a zip archive file with the same number of Python files, having the same names: `ex1.py`, `ex2.py`, ...[1]

Evaluation

The following command will be used to check the solutions:

```
python -m doctest ex1.py
```

Similarly for the other exercises. The solutions which pass all the tests will be considered as correct.

Exercise 1

Implement function `sumprod` which takes on input two n-sized tensor vectors $v, w \in \mathbb{R}^n$ and is defined as follows:

$$\texttt{sumprod}(v, w) = \sum_{i=1}^{n} v_i w_i \tag{1}$$

Hint: PyTorch actually provides an implementation of this function (under a different name...), but you can implement it yourself as well (and there are many possible implementations).

[1] In case you do not provide a solution for a particular exercise, no need to include it in the archive.

Exercise 2

Implement function `ata`[2] which takes as arguments two matrices:

- $M \in \mathbb{R}^{d \times d}$, a linear transformation matrix,
- $w \in \mathbb{R}^{n \times d}$, an input sentence (with one vector per word),

where n is the length of the sentence and d is the size of the vectors representing the individual words w_i in w. Function `ata` should apply M to each word vector w_i in sentence w. Put formally, $y = \texttt{ata}(M, w)$ should satisfy:

$$\forall_{i=1}^{n} y_i = M w_i \tag{2}$$

where $M w_i$ is the product of matrix M and word vector w_i.

Note: this exercise can be solved by directly implementing Eq. 3. While this solution will pass all the tests, it's not the most efficient (nor the most concise) one.

Exercise 3 (optional)

Extension of Ex. 2. Implement function `atab` which takes as arguments two matrices and one *bias vector*.

- $M \in \mathbb{R}^{d' \times d}$, a linear transformation matrix,
- $b \in \mathbb{R}^{d'}$, a bias vector,
- $w \in \mathbb{R}^{n \times d}$, an input sentence (with one vector per word),

Function `atab` should apply M to each word vector w_i in sentence w and add the bias vector b to the result. Put formally, $y = \texttt{atab}(M, b, w)$ should satisfy:

$$\forall_{i=1}^{n} y_i = M w_i + b \tag{3}$$

[2] For *apply to all*