Intro to DL Y2021 Semester B

Project part 3

Submission

Submission is in pairs or singles.

Data format and description

The data appear in moodle in two files:

- training_ex3_dl2021b.csv
- test_ex3_dl2021b.csv

The training data contains row ID, text data column, and a label (0 or 1):

id	sentence	**label**
1	My cat is fat.	1
…	…	…

The test data contains row ID, text data column, and no label:

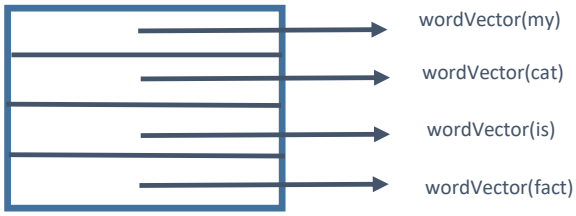
id	sentence
1	My dog is not fat.
…	…

The task

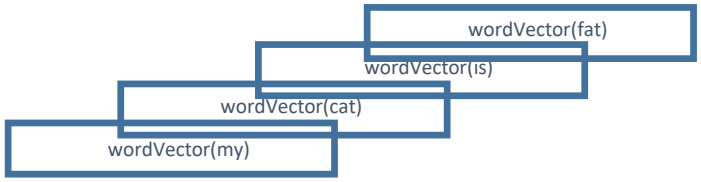
Your goal is to **predict the label** for test data values, based on the text (do not use the ID column! If you do, your score will be 0, and you will be on the bottom of the kaggle table).

1. Split your training data to the train part P1, and the test part P2 with sklearn.model_selection.train_test_split. Split percentages are up to you.
2. Preprocess your text data using nltk python package:
    a. Split your text data to words.
    b. You may choose to ignore some words; it is up to you what to keep.
    c. Compute matrix representation of each text for a 2D or 3D model as follows:

i. 1D (2d matrix model):
   1. There is a row for each word, in the same order they appear in the sentence.
   2. Every row is the word vector of that word:



ii. 2D (3d tensor model):
   1. Each word is a nx1 matrix, n is the size of your word vectors.
   2. Together, they form a 3d tensor:



3. Define a CNN1D or a CNN2D model keras.
   a. Use any number of layers you want.
   b. The last layer should have 1 neuron, your loss should be **binary_crossentropy**, and your metric should be **accuracy**.
4. Evaluate your model on the test set and save the label you produce for every item in .csv file, as described below.

<u>Note</u>: you can use any of the normalization and data analysis methods in sklearn to improve your scores.

Result submission

Your result should include item IDs from the test set and predicted label, and to be saved as csv file:

id	label
1	1
2	0
…	…

How kaggle works (a reminder)

Your results will be compared with the actual test dataset labels, and the resulting accuracy will be reported on the scoreboard of the competition. Note that public scoreboard will show

accuracy on 50% of the test set, and private (i.e., my) scoreboard will show accuracy on the whole test set. The final scoreboard will be published after submission & code checking is over, and your grade will be determined by your place in the competition.

Code submission

Submit your code on moodle, as a single <id1>_<id2>.py file (do not submit python notebooks!).

Note of warning: all code will be automatically checked for copying. If cheating is discovered, you will get grade 0 automatically and go on to face the scholarly committee.

How to boost your results (ideas):

1. Filter your data! Look at 'bad' sentences: are they 0 or 1? Measure their informativeness using your intuition: maybe, count the # of words. Maybe, dismiss numbers and stop-words (https://www.ranks.nl/stopwords)?
2. Whatever additional data about a single word you obtain, add it to the end of your word vectors as additional vector dimension. If you have doubt on how to encode it, use 1-hot representation.

3. If you have a sentence-level data, encode it in an additional vector (or 2, or 3,…) and incorporate it into your CNN-ready matrix (2d example shown):