Intro to DL Y2021 Semester B

Project part 4

Submission

Submission is in pairs or singles.

Data format and description

The data appear in moodle in three files:

- training_ex4_dl2021b.csv – training data, labeled
- test_ex4_dl2021b.csv – test data, unlabeled
- sampleSubmission_4.csv – sample submission file

The training data contains row ID, text data column, and a label (0,1, or 2):

id	sentence	**label**
1	My cat is fat.	2
…	…	…

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 OR spacy python package (more difficult to work with, better results):
    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 2D array representation of each text as follows:
        i. There is a vector for each word, in the same order they appear in the sentence.
        ii. The 2D sentence vector of a sentence includes word vectors in the order of words in the sentence:

sentenceVector=[wordVector(My), wordVector(cat), wordVector(is),…]

3. Define a recurrent neural model using keras LSTM or GRU layers, Bidirectional() is an option.
    a. Start with 1 layer and increase gradually, note that training is much slower for RNN than for other models.
    b. RNN is prone to overfitting, LSTM more so than GRU, so use batch training and dropout (do not be afraid of high dropout values).
    c. Use fully connected layers at the end, not too many of them, because they cannot learn better than RNN.
    d. The last layer should have 1 neuron, your loss should be **binary_crossentropy**, and your metric should be **accuracy**. Activation of the last neuron should be **sigmoid**.
4. Evaluate your model on the test set and save the label you produce for every item in .csv file, as described below.

Note: 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.