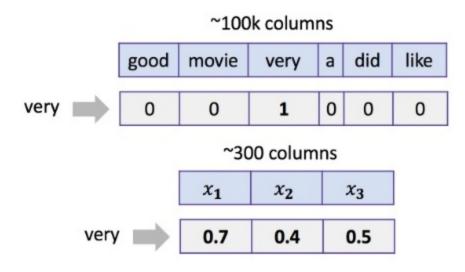




Congratulations! You passed!

points

Let's recall how we treated words as one-hot sparse vectors in BOW and dense embeddings in 1. neural networks:



Choose correct statements below.

Linear model on top of a sum of neural representations can work faster than on top of BOW.

Correct

This is true! We only need to train 300 parameters here. Don't forget to normalize these features row-wise!

For both word representations we can take a sum of vectors corresponding to tokens of any text to obtain good features for this text for further usage in linear model.

Correct

Yes, this is true. Don't forget to normalize these features row-wise!

You can replace word2vec embeddings with any random vectors to get a good features descriptor as a sum of vectors corresponding to all text tokens.

Un-selected is correct

For both word representations we can take a weighted sum of vectors corresponding to tokens of any text to obtain good features for this text for further usage in linear model. The weight for any token can be an IDF value for that token.

Correct

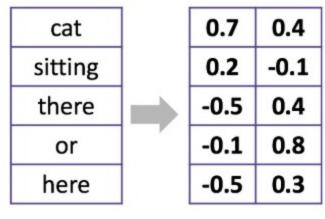
Yes, this is true. For BOW we effectively get bag of TF-IDF values, where TF is a binary variable. Don't forget to normalize these features row-wise!



Let's recall 1D convolutions for words: 2.

points

Word embeddings



What is the result of 1D convolution + maximum pooling over time for the following kernel without padding?

| 1 | 0 | |
|-----|---|--|
| 0 | 1 | |
| | | |
| 0.6 | | |

Correct Response

That's it!



points

3. Let's recall 1D convolutions for characters. Choose correct statements.



Un-selected is correct

1D convolutions for characters consume one-hot encoded vectors for characters.

That's right, they are not that long, so this is okay.



Correct This is true.