



List

Loupe Copy

Arrow Left
Sequence Models

Natural Language Processing & Word Embeddings

Graded Quiz • 30 min
Due Aug 19, 12:59 PM +06
Introduction to Word
Embeddings

Check

Congratulations! You passed!
To Pass 80% or higher

Natural Language Processing & Word
Embeddings

Practice questions

Grade

Completed

100%

Latest Submission Grade

Quiz: Natural Language

Processing & Word Embeddings

10 questions

Natural Language Processing & Word Embeddings

Programming
1. assignments

Suppose you learn a word embedding for a vocabulary of 10000 words. Then the embedding vectors should be 10000 dimensional, so as to capture the full range of variation and meaning in those words.

1 / 1 point

True

Quiz • 30 min

False

Natural Language Processing & Word Embeddings

Check

Correct

The dimension of word vectors is usually smaller than the size of the vocabulary. Most common sizes for word vectors ranges between 50 and 400.

Attempts 3 every 8 hours

2.

Try again

What is t-SNE?

1 / 1 point

Recheck Grade

To Pass 80% or higher

A linear transformation that allows us to solve analogies on word vectors

Grade

100%

A non-linear dimensionality reduction technique

We keep your highest score

A supervised learning algorithm for learning word embeddings

Thumbs Up

Thumbs Down

Sql

An open-source sequence modeling library

Check

Correct

Yes

3. Suppose you download a pre-trained word embedding which has been trained on a huge corpus of text. You then use this word embedding to train an RNN for a language task of recognizing if someone is happy from a short snippet of text, using a small training set.

x (input text)	y (happy?)
I'm feeling wonderful today!	1
I'm bummed my cat is ill.	0
Really enjoying this!	1

Then even if the word “ecstatic” does not appear in your small training set, your RNN might reasonably be expected to recognize “I’m ecstatic” as deserving a label $y = 1$.

1 / 1 point

True

False

Check

Correct

Yes, word vectors empower your model with an incredible ability to generalize. The vector for "ecstatic" would contain a positive/happy connotation which will probably make your model classify the sentence as a "1".

4.

Which of these equations do you think should hold for a good word embedding? (Check all that apply)

1 / 1 point

$$e_{boy} - e_{girl} \approx e_{brother} - e_{sister}$$

Check

Correct

Yes!

$$e_{boy} - e_{girl} \approx e_{sister} - e_{brother}$$

$$e_{boy} - e_{brother} \approx e_{girl} - e_{sister}$$

Check

Correct

Yes!

$$e_{boy} - e_{brother} \approx e_{sister} - e_{girl}$$

5.

Let E be an embedding matrix, and let o_{1234} be a one-hot vector corresponding to word 1234. Then to get the embedding of word 1234, why don't we call $E * o_{1234}$ in Python?

1 / 1 point

It is computationally wasteful.

The correct formula is $E^T * o_{1234}$.

This doesn't handle unknown words (<UNK>).

None of the above: calling the Python snippet as described above is fine.

Check

Correct

Yes, the element-wise multiplication will be extremely inefficient.

6.

When learning word embeddings, we create an artificial task of estimating $P(\text{target} \mid \text{context})$. It is okay if we do poorly on this artificial prediction task; the more important by-product of this task is that we learn a useful set of word embeddings.

1 / 1 point

True

False

Check

Correct

7.

In the word2vec algorithm, you estimate $P(t \mid c)$, where t is the target word and c is a context word. How are t and c chosen from the training set? Pick the best answer.

1 / 1 point

c and t are chosen to be nearby words.

c is the one word that comes immediately before t .

c is the sequence of all the words in the sentence before t .

c is a sequence of several words immediately before t .

Check

Correct

8.
Suppose you have a 10000 word vocabulary, and are learning 500-dimensional word embeddings. The word2vec model uses the following softmax function:

$$P(t \mid c) = \frac{e^{\theta_t^T e_c}}{\sum_{t'=1}^{10000} e^{\theta_{t'}^T e_c}}$$

Which of these statements are correct? Check all that apply.

1 / 1 point

θ_t and e_c are both 500 dimensional vectors.

Check

Correct

θ_t and e_c are both 10000 dimensional vectors.

θ_t and e_c are both trained with an optimization algorithm such as Adam or gradient descent.

Check

Correct

After training, we should expect θ_t to be very close to e_c when t and c are the same word.

9.

Suppose you have a 10000 word vocabulary, and are learning 500-dimensional word embeddings. The GloVe model minimizes this objective:

$$\min \sum_{i=1}^{10,000} \sum_{j=1}^{10,000} f(X_{ij})(\theta_i^T e_j + b_i + b_j' - \log X_{ij})^2$$

Which of these statements are correct? Check all that apply.

1 / 1 point

θ_i and e_j should be initialized to 0 at the beginning of training.

θ_i and e_j should be initialized randomly at the beginning of training.

Check

Correct

X_{ij} is the number of times word i appears in the context of word j.

Check

Correct

The weighting function $f(\cdot)$ must satisfy $f(0) = 0$.

Check

Correct

The weighting function helps prevent learning only from extremely common word pairs. It is not necessary that it

satisfies this function.

10.

You have trained word embeddings using a text dataset of m_1 words. You are considering using these word embeddings for a language task, for which you have a separate labeled dataset of m_2 words. Keeping in mind that using word embeddings is a form of transfer learning, under which of these circumstance would you expect the word embeddings to be helpful?

1 / 1 point

$m_1 \gg m_2$

$m_1 \ll m_2$

Check

Correct