Congratulations! You passed!

Grade received 90% | Latest Submission Grade 90% | To pass 80% or higher

Go to next item

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

0 / 1 point

- False
- True

Expand

Incorrect
No, the dimension of word vectors is usually smaller than the size of the vocabulary. Most common sizes for word vectors range between 50 and 1000.

2. True/False: t-SNE is a linear transformation that allows us to solve analogies on word vectors.

1 / 1 point

- False
- True

Expand

Correct
tr-SNE is a non-linear dimensionality reduction technique.

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.

1 / 1 point

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

True/False: Then even if the word “upset” does not appear in your small training set, your RNN might reasonably be expected to recognize “I'm upset” as deserving a label y = 0.

- True
- False

Expand

Correct
Yes, word vectors empower your model with an incredible ability to generalize. The vector for “upset” would contain a negative/unhappy connotation which will probably make your model classify the sentence as a "0".

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

1 / 1 point

- [ ] $e_{boy} - e_{girl} \approx e_{sister} - e_{brother}$
- [x] $e_{boy} - e_{brother} \approx e_{girl} - e_{sister}$

Correct
Yes!

- [x] $e_{boy} - e_{girl} \approx e_{brother} - e_{sister}$

Correct
Yes!

Typesetting math: 100% $e_{sister} - e_{girl}$

Expand

Correct
Great, you got all the right answers.

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

- ○ This doesn't handle unknown words (<UNK>).
- ○ None of the above: calling the Python snippet as described above is fine.
- ◉ It is computationally wasteful.
- ○ The correct formula is $E^T * o_{1234}$

Expand

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

6. When learning word embeddings, we create an artificial task of estimating $P(target \mid context)$. 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

- ○ False
- ◉ True

Expand

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 is the sequence of all the words in the sentence before t
- ○ c is a sequence of several words immediately before t
- ◉ c and $$t$$ are chosen to be nearby words.
- ○ $$c$$ is the one word that comes immediately before $$t$$

Processing math: 100%

Expand

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.

- ☐ After training, we should expect θ_t to be very close to e_c when t and c are the same word.
- ☑ θ_t and e_c are both 500 dimensional vectors.

  Correct

- ☑ $$\theta_t$$ and $$e_c$$ are both trained with an optimization algorithm such as Adam or gradient descent.

Expand

Correct
Great, you got all the right answers.

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

1 / 1 point

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

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

False

True

Expand

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

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 circumstances would you expect the word embeddings to be helpful?

1 / 1 point

m_1 << m_2

m_1 >> $$m_2$$

Processing math: 100%

Expand

Correct