Congratulations! You passed!

Grade received 100% **Latest Submission Grade** 100% **To pass** 80% or higher

Go to next item

1. True/False: Suppose you learn a word embedding for a vocabulary of 20000 words. Then the embedding vectors could be 1000 dimensional, so as to capture the full range of variation and meaning in those words. **1 / 1 point**

- False
- True

Expand

Correct
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 non-linear dimensionality reduction technique. **1 / 1 point**

- True
- False

Expand

Correct
t-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?)
Having a great time!	1
I'm sad it's raining.	0
I'm feeling awesome!	1

Even if the word "wonderful" does not appear in your small training set, what label might be reasonably expected for the input text "I feel wonderful!"?

- y=1
- y=0

Expand

Correct
Yes, word vectors empower your model with an incredible ability to generalize. The vector for “wonderful” would contain a negative/unhappy 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**

- [x] $e_{man} - e_{woman} \approx e_{uncle} - e_{aunt}$

  Correct
  The order of words is correct in this analogy.

- [ ] $e_{man} - e_{woman} \approx e_{aunt} - e_{uncle}$
- [ ] $e_{man} - e_{aunt} \approx e_{woman} - e_{uncle}$
- [x] $e_{man} - e_{uncle} \approx e_{woman} - e_{aunt}$

  Correct
  The order of words is correct in this analogy.

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**

- ( ) None of the above: calling the Python snippet as described above is fine.
- ( ) This doesn’t handle unknown words (<UNK>).
- (•) 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 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
- c is the sequence of all the words in the sentence before t

Expand

Correct

8. Suppose you have a 10000 word vocabulary, and are learning 100-dimensional word embeddings. The word2vec model uses the following softmax function: **1 / 1 point**

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

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

- [x] θ_t and e_c are both 100 dimensional vectors.

Correct

- [ ] θ_t and e_c are both 10000 dimensional vectors.
- [x] θ_t and e_c are both trained with an optimization algorithm.

Correct
To review this concept watch the *Word2Vec* lecture.

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

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

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 s_1 words. You are considering using these word embeddings for a language task, for which you have a separate labeled dataset of s_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

- ○ $s_1 << s_2$
- ◉ $s_1 >> s_2$

Expand

Correct

s_1 should transfer to s_2