Try again once you are ready

Grade received 70% **Latest Submission Grade** 70% **To pass** 80% or higher

Try again

1. True/False: Suppose you learn a word embedding for a vocabulary of 60000 words. Then the embedding vectors could be 60000 dimensional, so as to capture the full range of variation and meaning in those words.

1 / 1 point

- (•) False
- ( ) True

Expand

Correct
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.

0 / 1 point

- (•) True
- ( ) False

Expand

Incorrect
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

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 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 $y = 1$.

- False
- True

Expand

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

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

Correct
Yes!

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

Correct
Yes!

Expand

Correct
Great, you got all the right answers.

5. Let A be an embedding matrix, and let o_{4567} be a one-hot vector corresponding to word 4567. Then to get the embedding of word 4567, why don't we call $A * o_{4567}$ in Python? **1 / 1 point**

- None of the answers are correct: calling the Python snippet as described above is fine.
- This doesn't handle unknown words (<UNK>).
- The correct formula is $A^T * o_{4567}$
- It is computationally wasteful.

Expand

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

6. When learning word embeddings, words are automatically generated along with the surrounding words. **0 / 1 point**

- False
- True

Expand

Incorrect
No, we pick a given word and try to predict its surrounding words or vice versa.

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 one word that comes immediately before t

- c is the sequence of all the words in the sentence before t
- c and t are chosen to be nearby words.
- c is a sequence of several words immediately 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**

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

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

- False
- True

Expand

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

9. Suppose you have a 10000 word vocabulary, and are learning 500-dimensional word embeddings.The GloVe model minimizes this objective: **0 / 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: θ_i and e_j should be initialized to 0 at the beginning of training.

- False
- True

Expand

Incorrect
No, $$\theta_i$$ and $$e_j$$ should be initialized randomly at the beginning of training.

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 $$s_2$$