Natural Language Processing & Word Embeddings

TOTAL POINTS 10

1.	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 point			
	○ True				
	False				
2.	What is t-SNE?	1 point			
	A linear transformation that allows us to solve analogies on word vectors				
	A non-linear dimensionality reduction technique				
	A supervised learning algorithm for learning word embeddings				
	An open-source sequence modeling library				
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 point			

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

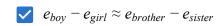
Then even if the word "ecstatic" does not appear in your small training set, your RNN might reasonably be expected to recognize "I'm ecstatic" as deserving a label y=1.

True

Fa	lse

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

1 point



- \square $e_{boy} e_{girl} \approx e_{sister} e_{brother}$
- $ightharpoonup e_{boy} e_{brother} \approx e_{girl} e_{sister}$
- $e_{boy} e_{brother} \approx e_{sister} e_{girl}$
- 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 point

- It is computationally wasteful.
- O 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.
- 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 point

- True
- False
- 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 point

- \bigcirc c is a sequence of several words immediately before t.
- \bigcirc c is the one word that comes immediately before t.
- c is the sequence of all the words in the sentence before t.

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

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.

- $\ensuremath{\checkmark}$ θ_t and e_c are both 500 dimensional vectors.
- θ_t and e_c are both 10000 dimensional vectors.
- $ightharpoonup heta_t$ and e_c are both trained with an optimization algorithm such as Adam or gradient descent.
- 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:

1 point

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

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

- θ_i and e_i should be initialized to 0 at the beginning of training.
- $oldsymbol{
 u}$ θ_i and e_j should be initialized randomly at the beginning of training.
- $\bigvee X_{ij}$ is the number of times word j appears in the context of word i.
- The weighting function f(.) must satisfy f(0) = 0.
- 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?



- $\bigcap m_1 \ll m_2$