1.	embed	se you learn a word embedding for a vocabulary of 10000 words. Then the ding vectors should be 10000 dimensional, so as to capture the full range of and meaning in those words.
		True
		False
		dimension of word vectors is usually smaller than the size of the vocabulary. common sizes for word vectors ranges between 50 and 400.
2.	What is	s t-SNE?
		A linear transformation that allows us to solve analogies on word vectors
		A non-linear dimensionality reduction technique
	Corre	ect
		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.

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

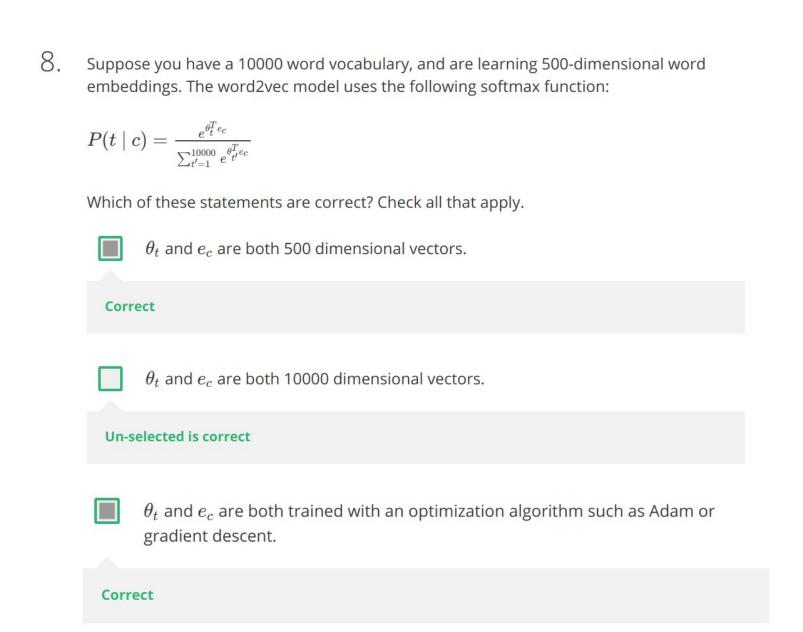
Correct

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 classified the sentence as a "1".

False

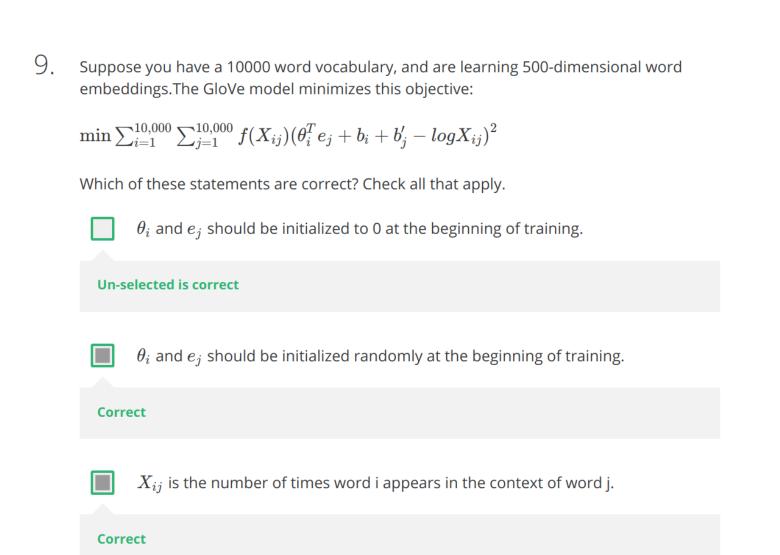
4.	Which of these equations do you think should hold for a good word embedding? (Check all that apply)
	$lacksquare$ $e_{boy} - e_{girl} pprox e_{brother} - e_{sister}$
	Correct Yes!
	$e_{boy} - e_{girl} pprox e_{sister} - e_{brother}$
	Un-selected is correct
	$lacksquare$ $e_{boy} - e_{brother} pprox e_{girl} - e_{sister}$
	Correct Yes!
	$e_{boy} - e_{brother} pprox e_{sister} - e_{girl}$
	Un-selected is correct
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?
	It is computationally wasteful.
	Correct Yes, the element-wise multiplication will be extremely inefficient.
	$igcup$ The correct formula is $E^T st o_{1234}.$
	This doesn't handle unknown words (<unk>).</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.		
	True		
	Correct		
	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.		
	$igcup_c$ and t are chosen to be nearby words.		
	Correct		
	igcap c is a sequence of several words immediately before $t.$		
	igcup c is the sequence of all the words in the sentence before $t.$		
	igcup c is the one word that comes immediately before $t.$		



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

Un-selected is correct





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?



 $m_1 >> m_2$

Correct

