Sequence Models

Week 2: Natural Language Processing and Word Embeddings

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.

Answer: True.

Comment: 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 Version 1: True/False: t-SNE is a linear transformation that allows us to solve analogies on word vectors.

Answer: False.

Comment: tr-SNE is a non-linear dimensionality reduction technique.

2 Version 2: What is t-SNE?

Answer: 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. 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!"?

Answer: y=1.

Comment: 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)

Answer: $e_{boy} - e_{girl} \approx e_{brother} - e_{sisiter}$;$e_{boy} - e_{brother} \approx e_{girl} - e_{sisiter}$.

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?

Answer: It is computationally wasteful.

Comment: The element-wise multiplication will be extremely inefficient.

6 When learning word embeddings, we pick a given word and try to predict its surrounding words or vice versa.

Answer: True.

Comment: Word embeddings are learned by picking a given word and trying to predict its surrounding words or vice versa.

7 In the word2vec algorithm, you estimate $P(t|c)$ 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.

Answer: c and t are chosen to be nearby words.

8 Suppose you have a 10000 word vocabulary, and are learning 100-dimensional word embeddings. The word2vec model uses the following softmax function: True/False: After training, we should expect θ_t to be very close to e_c when t and c are the same word.

Answer: False.

Comment: See lecture notes.

9 Suppose you have a 10000 word vocabulary, and are learning 500-dimensional word embeddings. The GloVe model minimizes this objective: True/False: X_{ij} is the number of times word j appears in the context of word i.

Answer: True.

Comment: 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?

Answer: $s_1 >> s_2$.

Comment: s_1 should transfer to s_2.