1) Which techniques can be utilized to assess the quality of the word embeddings generated by our model?

- ○ Evaluating semantic relatedness
- ○ Identifying synonyms
- ○ Solving analogy problems
- ◉ All of the above

Yes, the answer is correct.
Score: 1
Accepted Answers:
*All of the above*

1. **Evaluating semantic relatedness**: This involves measuring how closely word embeddings capture the meanings and relationships between words. Techniques like cosine similarity can be used here.
2. **Identifying synonyms**: Checking if synonyms are close to each other in the embedding space serves as an indicator of the quality of the embeddings.
3. **Solving analogy problems**: This tests the ability of embeddings to capture relationships (e.g., "king - man + woman = queen") and is a well-known method for evaluating word embeddings.

## Conclusion:

All these techniques effectively assess the quality of word embeddings, so the correct option is **All of the above.**

2) At the input layer of a continuous bag of words model, we multiply a one-hot vector $x \in \mathbb{R}^{|V|}$ with the parameter matrix $W \in \mathbb{R}^{k \times |V|}$. What does t each column of $W$ correspond to?

- ◉ the representation of the $i$-th word in the vocabulary
- ○ the $i$-th eigen vector of the co-occurrence matrix

Yes, the answer is correct.
Score: 1
Accepted Answers:
*the representation of the $i$-th word in the vocabulary*

## The representation of the i-th word in the vocabulary.

## Explanation:

- In this context, the matrix $W$ contains the word embeddings for each word in the vocabulary. Each column represents the latent vector (embedding) associated with a specific word, enabling the model to capture semantic meanings based on their contexts.

3) Suppose that we use the continuous bag of words (CBOW) model to find vector representations of words. Suppose further that we use a context window of size 3 (that is, given the 3 context words, predict the target word $P(w_t|(w_i, w_j, w_k))$). The size of word vectors (vector representation of words) is chosen to be 100 and the vocabulary contains 10,000 words. The input to the network is the one-hot encoding (also called 1-of-$V$ encoding) of word(s). How many parameters (weights), excluding bias, are there in $W_{\text{word}}$? Enter the answer in thousands. For example, if your answer is 50,000, then just enter 50.

1000

Yes, the answer is correct.
Score: 1
Accepted Answers:
*(Type: Numeric) 1000*

To calculate the number of parameters (weights) in $W_{\text{word}}$ for the Continuous Bag of Words (CBOW) model, we need to consider the following:

1. **Size of Word Vectors** ($k$): 100
2. **Vocabulary Size** ($|V|$): 10,000

The parameter matrix $W_{\text{word}}$ is of size $k \times |V|$. This means:

$$\text{Number of parameters} = k \times |V| = 100 \times 10,000 = 1,000,000$$

Now, to express this in thousands:

$$1,000,000 \div 1,000 = 1,000$$

## Final Answer:

**1,000**

4) You are given the one hot representation of two words below:
CAR= $[1, 0, 0, 0, 0]$, BUS= $[0, 0, 0, 1, 0]$

What is the Euclidean distance between CAR and BUS?

1.414

Yes, the answer is correct.
Score: 1
Accepted Answers:
*(Type: Range) 1.40,1.42*

1. **Define the vectors:**
   - $\text{CAR} = [1, 0, 0, 0, 0]$
   - $\text{BUS} = [0, 0, 0, 1, 0]$
2. **Use the Euclidean distance formula:**

$$d(\mathbf{u}, \mathbf{v}) = \sqrt{\sum (u_i - v_i)^2}$$

3. **Calculate the differences:**

$$\begin{gathered}(1-0)^2 = 1^2 = 1 \\ (0-0)^2 = 0^2 = 0 \\ (0-0)^2 = 0^2 = 0 \\ (0-1)^2 = (-1)^2 = 1 \\ (0-0)^2 = 0^2 = 0\end{gathered}$$

4. **Sum the squares:**

$$1 + 0 + 0 + 1 + 0 = 2$$

5. **Take the square root:**

$$d(\text{CAR}, \text{BUS}) = \sqrt{2}$$

5) Let $count(w, c)$ be the number of times the words $w$ and $c$ appear together in the corpus (i.e., occur within a window of few words around each ***1 poi*** other). Further, let $count(w)$ and $count(c)$ be the total number of times the word $w$ and $c$ appear in the corpus respectively and let $N$ be the total number o words in the corpus. The PMI between $w$ and $c$ is then given by:

- $\log \frac{count(w,c)*count(w)}{N*count(c)}$
- $\log \frac{count(w,c)*count(c)}{N*count(w)}$
- $\log \frac{count(w,c)*N}{count(w)*count(c)}$

Yes, the answer is correct.
Score: 1

Accepted Answers:
$\log \frac{count(w,c)*N}{count(w)*count(c)}$

The Pointwise Mutual Information (PMI) between two words $w$ and $c$ is defined as a measure of how much the occurrence of one word contributes to the occurrence of the other, compared to what would be expected if the two words were independent.

The correct formula for PMI is:

$$\text{PMI}(w, c) = \log \frac{\text{count}(w, c) \cdot N}{\text{count}(w) \cdot \text{count}(c)}$$

This can be expressed in the form of the options you provided. The correct option that matches the PMI formula is:

$$\log \frac{\text{count}(w,c) \cdot N}{\text{count}(w) \cdot \text{count}(c)}$$

This indicates how the co-occurrence of $w$ and $c$ compares to their individual frequencies, normalized by the total number of words $N$.

6) Suppose we are learning the representations of words using Glove representations. If we observe that the cosine similarity between two representations $v_i$ and $v_j$ for words `$i$' and `$j$' is very high. which of the following statements is true?( parameter $b_i$ = 0.02 and $b_j$ = 0.07)

- $X_{ij} = 0.02$
- $X_{ij} = 0.2$
- $X_{ij} = 0.88$
- $X_{ij} = 0$

Yes, the answer is correct.
Score: 1
Accepted Answers:
$X_{ij} = 0.88$

In GloVe (Global Vectors for Word Representation), the cosine similarity between two word vectors $v_i$ and $v_j$ is calculated as follows:

$$\text{cosine similarity}(v_i, v_j) = \frac{v_i \cdot v_j}{\|v_i\| \|v_j\|}$$

If the cosine similarity is very high, it indicates that the two word vectors $v_i$ and $v_j$ are very similar in direction and hence convey similar meanings.

Given that you are asking about the value of $X_{ij}$ (which usually represents the co-occurrence or some derived metric between $i$ and $j$), one would typically relate this to the differences in the biases $b_i$ and $b_j$.

In GloVe, the model learns to predict relationships based on the co-occurrence counts, which means higher similarity may correlate with specific values for $X_{ij}$.

Given the context and typical interpretations, if we have very high cosine similarity, it often implies that $X_{ij}$ would not be zero, as that would suggest no relationship.

## Based on the values provided:

- If $b_i = 0.02$ and $b_j = 0.07$:
  1. **If** $X_{ij} = 0$: This would imply no co-occurrence, which contradicts the high similarity.
  2. **If** $X_{ij} = 0.02$ or $X_{ij} = 0.2$: These values suggest some relationship, but without additional context about the distribution of values, it's hard to definitively choose between them.
  3. **If** $X_{ij} = 0.88$: This would represent a significant co-occurrence, aligning well with the indication of high similarity.

## Conclusion:

Given that the cosine similarity is very high, the most reasonable assumption would be that $X_{ij}$ reflects this strong relationship and is likely not zero.

If I had to choose based on the highest similarity, I would indicate that:

**$X_{ij} = 0.88$ would be the most appropriate choice.**

7) Which of the following is a disadvantage of one hot encoding?

- (●) It requires a large amount of memory to store the vectors
- ( ) It can result in a high-dimensional sparse representation
- ( ) It cannot capture the semantic similarity between words
- ( ) All of the above

No, the answer is incorrect.
Score: 0

Accepted Answers:
*All of the above*

1. **It requires a large amount of memory to store the vectors**: One-hot encoding creates a vector that is as long as the size of the vocabulary. This can lead to large memory usage if the vocabulary is extensive.
2. **It can result in a high-dimensional sparse representation**: The resulting vectors are sparse because most of the elements in a one-hot encoded vector are zeros, which can make computations inefficient.
3. **It cannot capture the semantic similarity between words**: One-hot encoding treats all words as being equidistant from each other since it does not capture any relationship or similarity between words. For example, "king" and "queen" would be as different as "king" and "apple".

## Conclusion:

Given these points, the correct answer is:

## All of the above.

9) Suppose we are learning the representations of words using Glove representations. If we observe that the cosine similarity between two representations $v_i$ and $v_j$ for words 'i' and 'j' is very high. which of the following statements is true?( parameter $b_i$ = 0.02 and $b_j$ = 0.05

- ( ) $X_{ij} = 0.03.$
- (●) $X_{ij} = 0.8.$
- ( ) $X_{ij} = 0.35.$
- ( ) $X_{ij} = 0.$

Yes, the answer is correct.
Score: 1
Accepted Answers:
$X_{ij} = 0.8.$

Since we observed that the cosine similarity is very high, it implies a strong relationship between the vectors $v_i$ and $v_j$. Thus, the most reasonable choice among the provided options would be:

$$X_{ij} = 0.8$$

This choice reflects a significant relationship that aligns with the high cosine similarity observed.

10) We add incorrect pairs into our corpus to maximize the probability of words that occur in the same context and minimize the probability of words that occur in different contexts. This technique is called-

- Hierarchical softmax
- Contrastive estimation
- Negative sampling
- Glove representations

Yes, the answer is correct.
Score: 1
Accepted Answers:
*Negative sampling*

**Negative sampling** is used in models like Word2Vec to efficiently train embeddings by focusing on a small number of negative examples (incorrect pairs) instead of trying to consider all possible pairs. The goal is to adjust the model such that it increases the probability of correct word-context pairs while decreasing the probability of incorrect pairs.

Here's why the other options do not fit:

- **Hierarchical softmax**: This is a method for reducing the computational burden in calculating softmax over a large vocabulary by organizing the output labels in a binary tree structure, but it doesn't specifically involve adding incorrect pairs.
- **Contrastive estimation**: While it relates to maximizing the likelihood of correct pairs versus incorrect ones, it's more general and not primarily focused on the framework that Word2Vec uses.
- **GloVe representations**: GloVe (Global Vectors for Word Representation) captures global statistical information but does not inherently involve the use of incorrect pairs in the same way that negative sampling does.

Thus, in the context of maximizing and minimizing probabilities of context pairs, **Negative sampling** is the correct term. Thank you for your patience!

**Question 8:**

Which of the following is an advantage of using the skip-gram method over the bag-of-words approach?

- The skip-gram method is faster to train
- **The skip-gram method performs better on rare words**
- The bag-of-words approach is more accurate
- The bag-of-words approach is better for short texts

**Question 9:**

What is the computational complexity of computing the softmax function in the output layer of a neural network?

- **O(n)**
- $O(n^2)$
- O(nlogn)
- O(logn)

**Question 10:**

How does Hierarchical Softmax reduce the computational complexity of computing the softmax function?

- It replaces the softmax function with a linear function
- **It uses a binary tree to approximate the softmax function**
- It uses a heuristic to compute the softmax function faster
- It does not reduce the computational complexity of computing the softmax function