4705-Homework 5

Xinyi Zhao

November 16, 2023

Negative Sampling

With negative sampling, when we have that $w_{c_1}, w_{c+1}, w_c)$ is a positive sample, we use sigmoid to approximate

$$p(w_c|w_{c-1}, w_{c+1})\ \sigma(b_{w_c}^T a_{avg} = \frac{1}{1 + exp(-b_{w_k}^T a_{avg})}$$

similarly we sample K words w_k that are not in the context,

$$p(w_c|w_{c-1}, w_{c+1})\ 1 - \sigma(b_{w_c}^T a_{avg} = \sigma(-b_{w_k}^T a_{avg}) = \frac{1}{1 + exp(b_{w_k}^T a_{avg})}$$

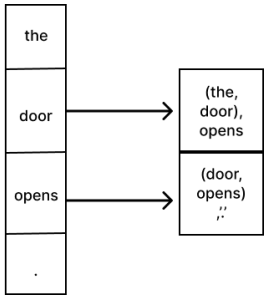
Thus we get:

$$p(w_c|w_{c-1}.w_{c+1}) = \frac{1}{1 + exp(-b_{w_c}^T a_{avg})} E_{w_k\ p_{sample}(w)}(\Pi_{k=1}^k \frac{1}{1 + exp(b_{w_k}^T a_{avg})}$$

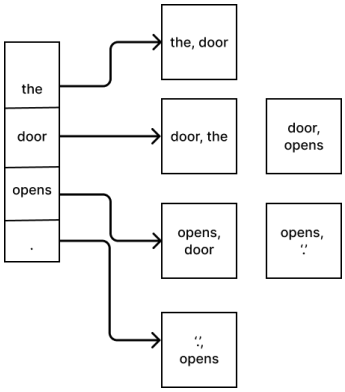
1

Let w be some word in the vocabulary V and let e_w be it's one-hot encoding (pretend the word is actually integer w, we might have itos[w] = "cat" for example depending on how we set up the hash map between words and integers). Explain why $B^T e_w = b_w \in R^d$ and why this multiplication selects the wth column of B^T. Remember, if $B \in R^{|V| \times d}$ then $B^T \in R^{d \times |V|}$.

Response: The one-hot encoding of a word w is vector e_w where all elements are 0, except for the w-th element which is 1. And B^T is a matrix



(a) illustration of cbow



(b) illustration of skip gram

with shape $d \times |V|$. The matrix multiplication:

$$B^T e_w = \begin{pmatrix} b_1 & b_2 & \cdots & b_{|V|} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} = 0b_1 + 0b_2 + \cdots + 1b_w + \cdots + 0b_{|V|} = b_w$$

Thus, $B^T e_w = b_w \in \mathbb{R}^d$, where b_w is the w.

2

Assume you do CBOW and Skip-Gram with negative sampling. As- sume m = 1. Which method, on average, will get more training sam- ples? Suppose there are 3 sentences with 7, 8, and 11 tokens. How many training sampling (positive training samples), will each method get. Draw a picture of a sentence with token counts and think about the number of samples each method gives. This is why Skip-Gram is used more often. It is more "sample efficient": you get more training data per Corpus.

Response: In CBOW, we generate one sample using the immediate surround words(1 left and 1 right) so in a sentence with n tokens with can generate n-2 samples. For each target word, it predicts the surrounding context words(1 before and 1 after).

So: 7 tokens: CBOW: 5 training samples. Skip-Gram: Each word generates 2 samples, but the first and last words will only generate 1 sample each. So 12 samples.

8 tokens: CBOW: 6 training samples. Skip-Gram: 14 training samples.

11 tokens: CBOW: 9 training samples Skip-Gram: 20 training samples.

3

Derivative with Respect to b_{w_o}

For the derivative with respect to b_w, the first term is the derivative of $-b_{w_o}^T a_{w_c}$, which is $-a_{w_c}$,

For the second term, only when $b_w = b_{w_o}$ is the Numerator $! = 0$. So the second derivative is

$$\frac{\exp(b_{w_o}^T a_{w_c}) \cdot a_{w_c}}{\sum_{w \in V} \exp(b_w^T a_{w_c})}$$

So the gradient with respect to b_{w_o} is

$$\frac{\partial L}{\partial b_{w_o}} = -a_{w_c} + \frac{\exp(b_{w_o}^T a_{w_c}) \cdot a_{w_c}}{\sum_{w \in V} \exp(b_w^T a_{w_c})}$$

Derivative with Respect to a_{w_c}

The first term is the derivative of $-b_{w_o}^T a_{w_c}$ with respect to a_{w_c}, which is simply $-b_{w_o}$.

Applying the chain rule, the second term becomes:

$$\frac{\sum_{w \in V} \exp(b_w^T a_{w_c}) \cdot b_w}{\sum_{w \in V} \exp(b_w^T a_{w_c})}$$

So, the gradient with respect to a_{w_c} is:

$$\frac{\partial L}{\partial a_{w_c}} = -b_{w_o} + \frac{\sum_{w \in V} \exp(b_w^T a_{w_c}) \cdot b_w}{\sum_{w \in V} \exp(b_w^T a_{w_c})}$$

Explanation

$-b_{w_o}$ represents the hard guess. It is the output vector corresponding to the actual target word with the context word w_c.

$\sum_{w \in V} \frac{log \sum_{w \in V} expb_w^T a_{w_c}}{\sum_{u \in V} expb_u^T a_{w_c}}$is the "expected value" because it's a weighted average of all output vectors in the vocabulary, where the weights are the exponential terms of the dot products between a_{w_c} and each b_u.