Training a Word Embedding Model

Setup

Phrase: _____ _____ _____ _____

# of words: 4 Context size: 1
# of dimensions: 2 Learning rate: 0.1

Context words

Training pairs

	(centre)		(left)	(right)
(first)	_____	:		_____
(second)	_____	:	_____	_____
(third)	_____	:	_____	_____
(fourth)	_____	:	_____	

Random numbers between 0 and 2

Centre word embeddings (v)

Word	Dimension 1 (X)	Dimension 2 (Y)

Random numbers between 0 and 2

Context embeddings (u)

Word	Dimension 1 (X)	Dimension 2 (Y)

Training

1. Choose Training Example

Pick one centre word and an associated context word.

(centre) (context)

_____ _____

Make sure that the words are next to each other in the beginning phrase!

Now, write down the initial embeddings for each word.

v_w → (centre): _____ _____

u_c → (context): _____ _____

Centre embeddings come from the first matrix, while context embeddings come from the second.

2. Compute Similarity

Next, take the **dot product** between the centre words and all adjacent context words.

similarity (*dot product*) = $(v_{wx} * u_{cx}) + (v_{wy} * u_{cy})$

Use the following formula, where u_{cx} and u_{cy} are the X and Y values of the context word, and v_{wx} and v_{wy} are the X and Y values of the centre word. Write the output below:

_____ _____

If the word is on either of the edges, it will only have one adjacent context word.

Then, take the exponential of each of those values ($e^{similarity}$)

_____ _____

3. Compute Softmax

Take the sum of the previously calculated values:

_____

And then divide the similarity values by the sum you just took:

_____ _____

Congratulations! You now have your model's predicted probabilities, in decimal form. You can check the percentage predictions by multiplying that by 100.

_____

4. Improve the Embeddings

First, we want to compute the loss. The loss measures how wrong the model's predictions are.

$$Cross\text{-}entropy = (p(x) * -log(q(x)))$$

That looks very scary, but all it's doing is:

- taking the logarithm of the previous, decimal prediction
- multiplying it by 0 if it's not the chosen ("true) context, or multiplying it by 1, if it is the true context word.

The logarithm is used in this scenario, in order to punish incorrect answers harder if they're more sure of themself. Perform the loss function on the softmax value belonging to the "true" context word.

_____

The next part of training is figuring out what changes need to be made to the embeddings. Use the following formula:

$$\underset{\text{true context embedding}}{\_\_\_\_\_} - [(\underset{\text{true context word softmax}}{\_\_\_\_\_} * \underset{\text{Context X 1}}{\_\_\_\_\_} + \underset{\text{true context word softmax}}{\_\_\_\_\_} * \underset{\text{Context X 2}}{\_\_\_\_\_}),$$
$$(\underset{\text{true context word softmax}}{\_\_\_\_\_} * \underset{\text{Context Y 1}}{\_\_\_\_\_} + \underset{\text{true context word softmax}}{\_\_\_\_\_} * \underset{\text{Context Y 2}}{\_\_\_\_\_})]$$

Finally, multiply the centre embedding by the softmax of the true context word, multiply it by the learning rate(0.1), and add it to the embedding.

$$\underset{\text{New centre embedding}}{\_\_\_\_\_} = \underset{\text{Centre embedding}}{\_\_\_\_\_} + (\underset{\text{Context word softmax}}{\_\_\_\_\_} * \underset{\text{Centre embedding}}{\_\_\_\_\_} * 0.1)$$