Hide menu

Lecture: Word Embeddings

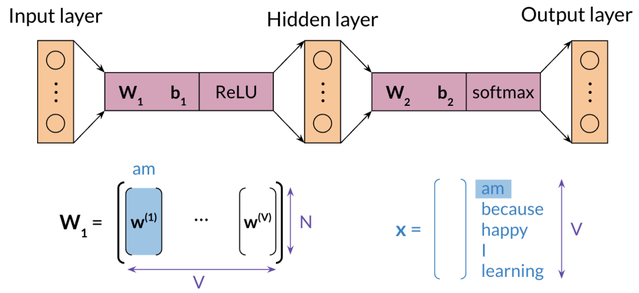
- **Video:** Week Introduction 1 min
- **Video:** Overview 2 min
- **Reading:** Overview
- **Video:** Basic Word Representations 3 min
- **Reading:** Basic Word Representations 5 min
- **Video:** Word Embeddings 3 min
- **Reading:** Word Embeddings 4 min
- **Video:** How to Create Word Embeddings 3 min
- **Reading:** How to Create Word Embeddings? 4 min
- **Video:** Word Embedding Methods 3 min
- **Reading:** Word Embedding Methods 4 min
- **Video:** Continuous Bag-of-Words Model 4 min
- **Reading:** Continuous Bag of Words Model 3 min
- **Video:** Cleaning and Tokenization 4 min
- **Reading:** Cleaning and Tokenization 5 min
- **Video:** Sliding Window of Words in Python 3 min
- **Reading:** Sliding Window of words in Python 10 min
- **Video:** Transforming Words into Vectors 3 min
- **Reading:** Transforming Words into Vectors 2 min
- **Lab:** Lecture Notebook - Data Preparation 30 min
- **Video:** Architecture of the CBOW Model 3 min
- **Reading:** Architecture for the CBOW Model 4 min
- **Video:** Architecture of the CBOW

Week 4 > Extracting Word Embedding Vectors

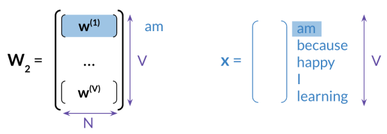
Previous Next

Extracting Word Embedding Vectors

There are two options to extract word embeddings after training the continuous bag of words model. You can use $w1$ as follows:



If you were to use $w1$, each column will correspond to the embeddings of a specific word. You can also use $w2$ as follows:



The final option is to take an average of both matrices as follows:

$$W_3 = 0.5\,(W_1 + W_2^T) = \begin{bmatrix} w_3^{(1)} & \cdots & w_3^{(V)} \end{bmatrix}_{V \times N}$$

Mark as completed

Like Dislike Report an issue