An Introduction to Word2Vec in NLP

An intuitive mathematical explanation to Word2Vec

Sai Pavan Yekula · Follow

Published in Towards Data Science

5 min read · Apr 3, 2022

Photo by Sven Brandsma on Unsplash

> *The tailor showed her how to sew a **button** onto her jacket. Meanwhile, her friend waited outside looking at the elevator **buttons.***

As humans, it is instinctive to notice the different meanings of the word “Button” in the above two sentences. But how can a machine learning model achieve this?

The first task of anyone working with Natural Language is to represent words as numbers. Many techniques such as One-Hot encoding, TF-IDF,N-grams have been used over the past decade. But these techniques fail to incorporate word semantics and are often sparse representations. This article introduces a word embedding technique called Word2Vec.

Word2Vec

> *“You shall know a word by the company it keeps” — John Rupert Firth*

Word2Vec is a state of the art algorithm to generate fixed length distributed vector representation of all the words in huge corpus. The effectiveness of Word2Vec is due two reasons — One, the use of fixed size vectors which means the vector size does not depend on the number of unique words in the corpus. Second, incorporating semantic information in the vector representations. Word2Vec vectors are highly efficient at grouping similar words together. The algorithm can make strong estimates based on the position of the word in the corpus. For example, ”Kid” and “Child” are similar and hence their vector representation will be very similar.

Word2Vec can be implemented in two architectures — **Continuous Bag of Words(CBOW) and Skip-Gram.** The main idea of Word2Vec revolves around predicting the context (outside) words based on a center word or vice versa in a fixed size window.

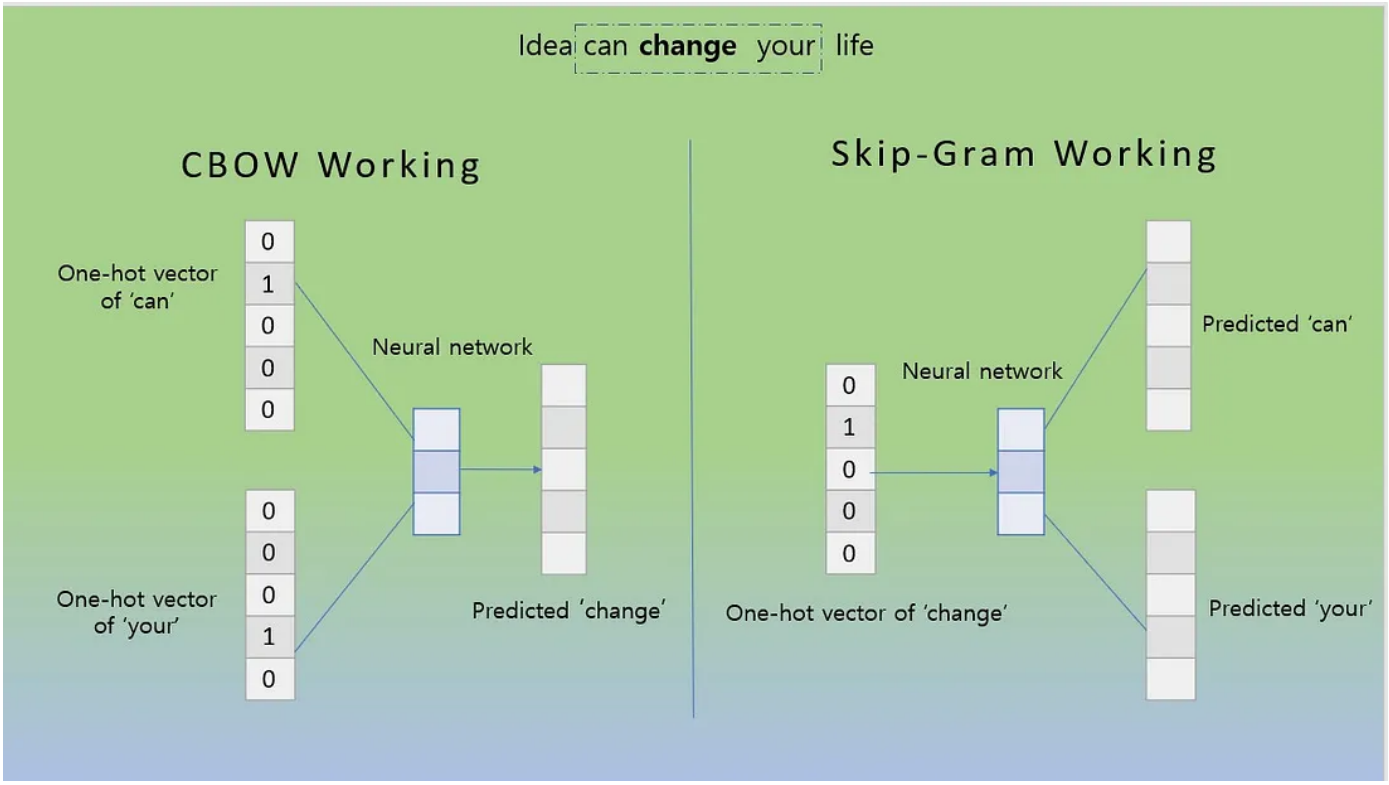
For example , consider the following part of a corpus :

> …. possibility of having a dream **come** true that makes life interesting…..

In the above example, with a fixed window of 3, let “***come***” be the center word and “***dream***” and “***true***” be the outside words. CBOW predicts the probability of *dream* and *true* given the center word *come* and Skip-Gram predicts the center word come, given the context words *dream* and *true*.

- CBOW — Predicts the center word based on the context (outside) words.

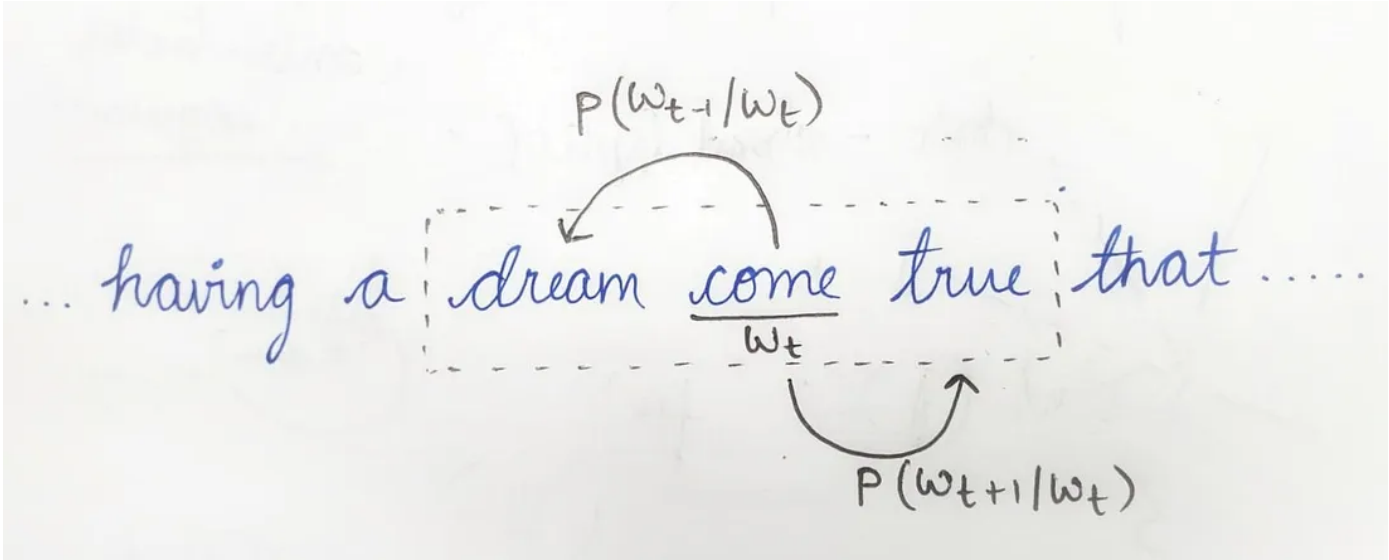
- Skip-Gram — Predicts the context words based on the center word



Architecture of CBOW and Skip Gram. Example sentence — "Idea can change your life". Image by author.

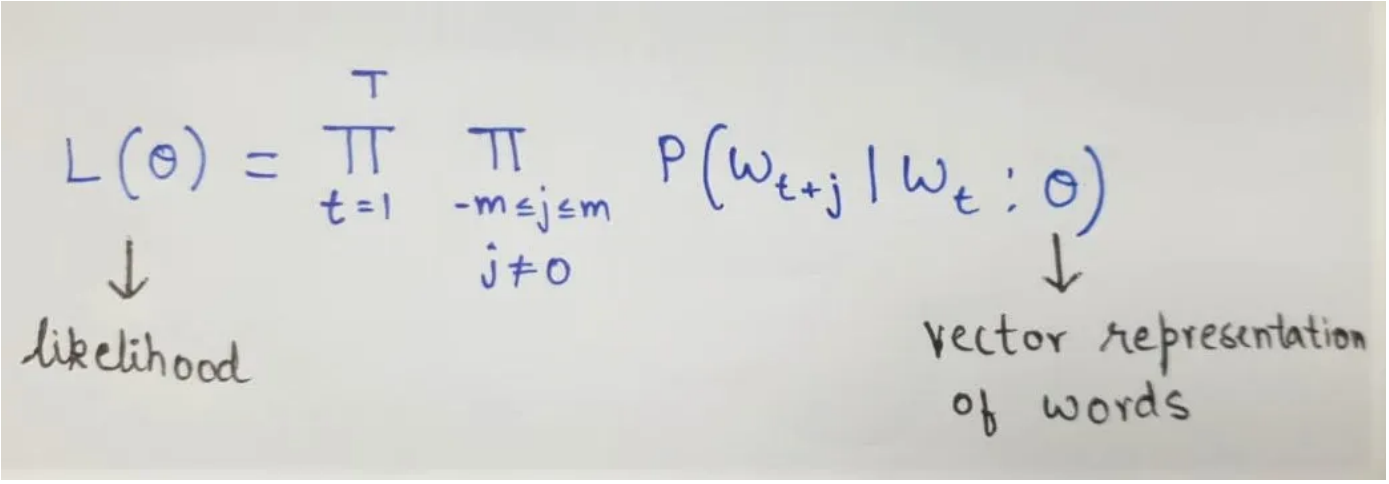
Objective Function

Objective function also called as error or cost function. During backpropagation, the neural network computes the minimum of the objective function using gradient descent. Below is an example representation of probability of each word in a fixed size window.



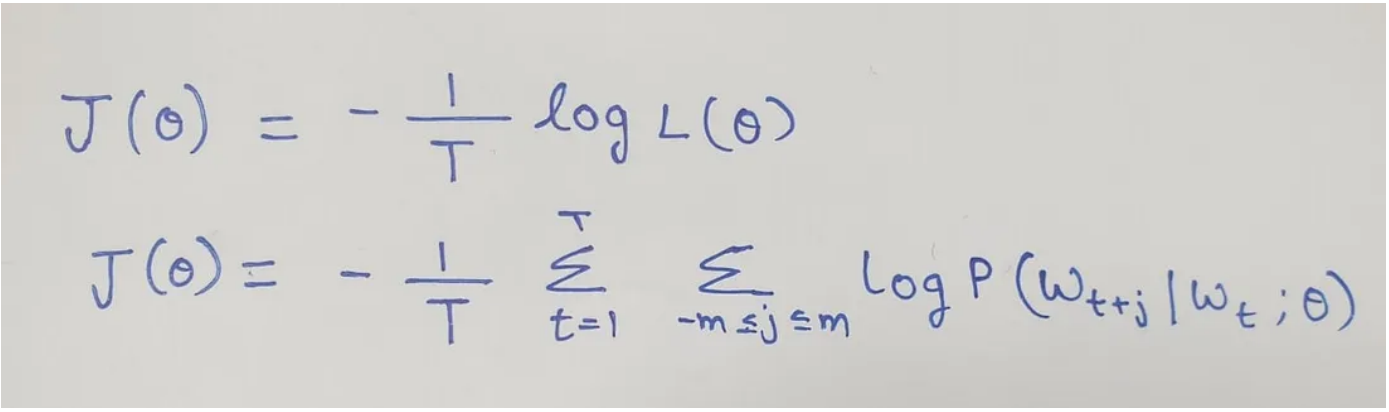
Example window of size 1. Context words are predicted using the center word “come”. Image by author.

Consider a corpus of size T. Given a center word W_t, context words within a fixed window size of m is predicted for each position of t (1,2…,T). The likelihood is given by:



Likelihood function. Image by author.

Likelihood function depends on the parameter θ. θ is all the variables to be optimized and it is the vector representation of word. The objective function (also called cost function or loss function) is the average negative of the log likelihood.



Objective function (Loss function)- Average Negative Log Loss. Image by author.

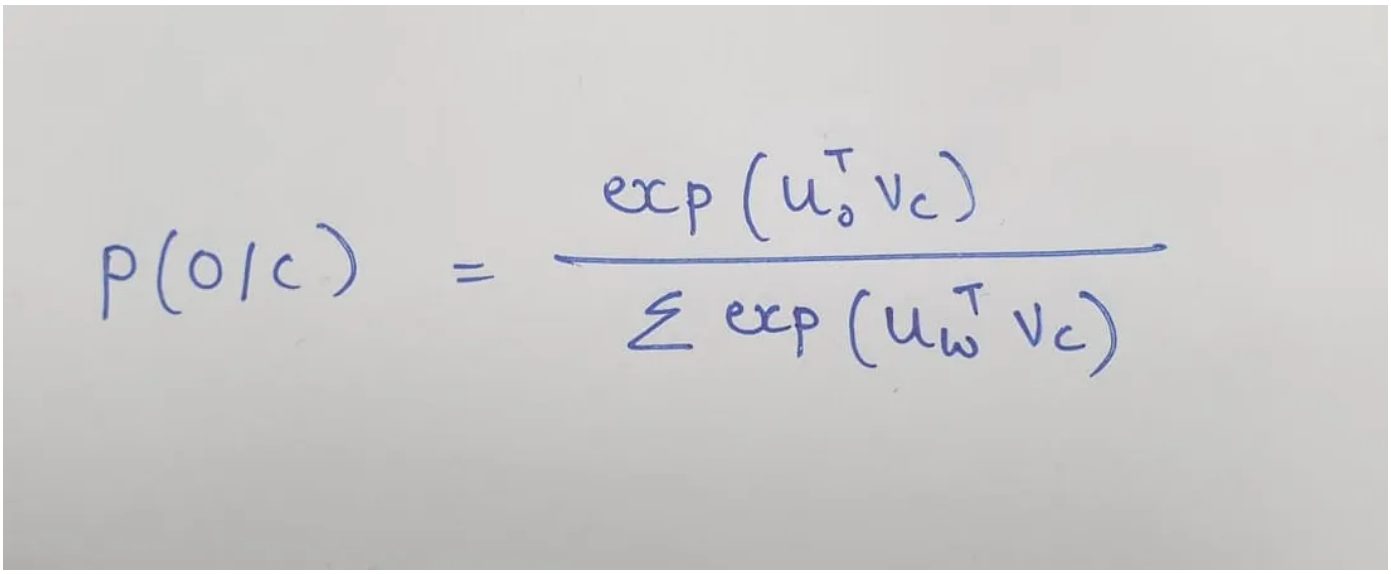
The function averages of over all the center words and hence the function doesn't depend on the number of unique words in the corpus. Objective function is minimized using an approach called gradient descent.

How to Calculate $P(W_{t+J} / W_t)$?

Each word has two vectors :

- v when the word is a center word
- u when the word is a context(outside) word

For a center word c and context word o,



Probability of outside word o given the center word c. Image by author.

Since we need positive values, exponential is used. The dot product gives the similarity of o and c, in this case larger dot product implies larger probability. The denominator normalizes over the entire corpus to give a probability distribution i.e., the numerator is divided over the similarity of ever word in the corpus. In fact, the above function is a SoftMax function and it makes sense because:

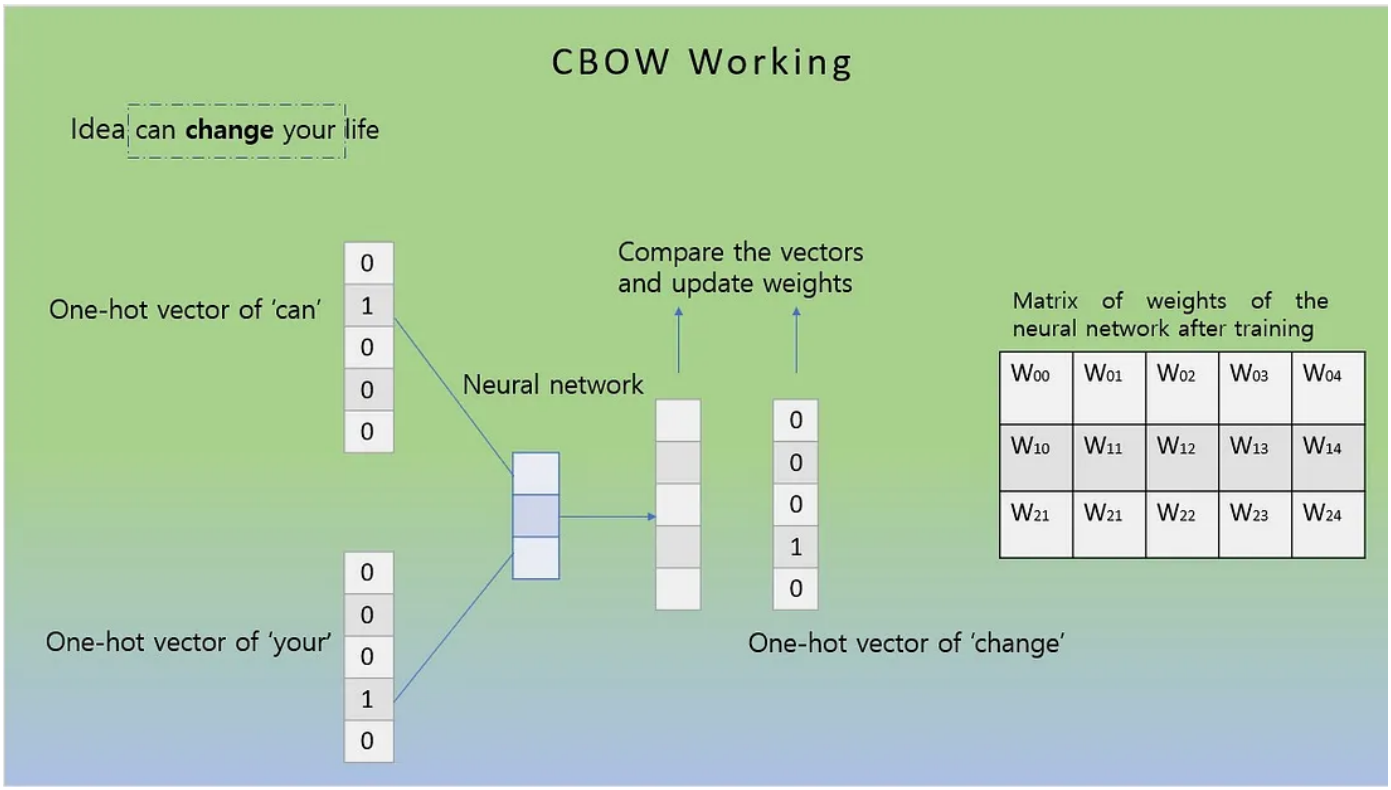
- Max: Amplifies the probability of the larges probable variable.
- Soft: Assigns some value to even the smallest probable variables.

We want the context words that surely come with the center word i.e., with high probability. Hence SoftMax function is the way to go.

How are the final Vectors formed?

After training the neural network with the above objective function, weights are obtained for each window. These weights are stored in a matrix. To obtain the final

dense vectors of each words, the weight matrix is multiplied by the corresponding One-Hot vector of each word.



A matrix of weights of the trained neural network is obtained. Image by author.

Conclusion

Word embedding is crucial step in Natural Language Processing. Obtaining vector representations of words using Word2Vec is a highly efficient because the vectors so formed are dense and carry sematic information which is crucial to any NLP application.

Thanks for reading this article! Leave a comment below if you have any questions. Be sure to follow me to get updates about my latest medium posts on machine learning and data science :) .You can reach out to me on LinkedIn if you have any questions.

NLP Naturallanguageprocessing Machine Learning Deep Learning

Word Embeddings

Follow

Written by Sai Pavan Yekula

18 Followers · Writer for Towards Data Science

Data Science & Machine Learning Enthusiast. I love to learn about the Mathematics behind Machine Learning. I believe ML and Data Science are Maths in disguise:)

More from Sai Pavan Yekula and Towards Data Science

Sai Pavan Yekula

K-Nearest Neighbors(KNN)-A Lazy Algorithm

K-Nearest Neighbors is one of the fundamental supervised machine learning algorithms. It is an essential instance based algorithm used for...

3 min read · Dec 19, 2021

10 1

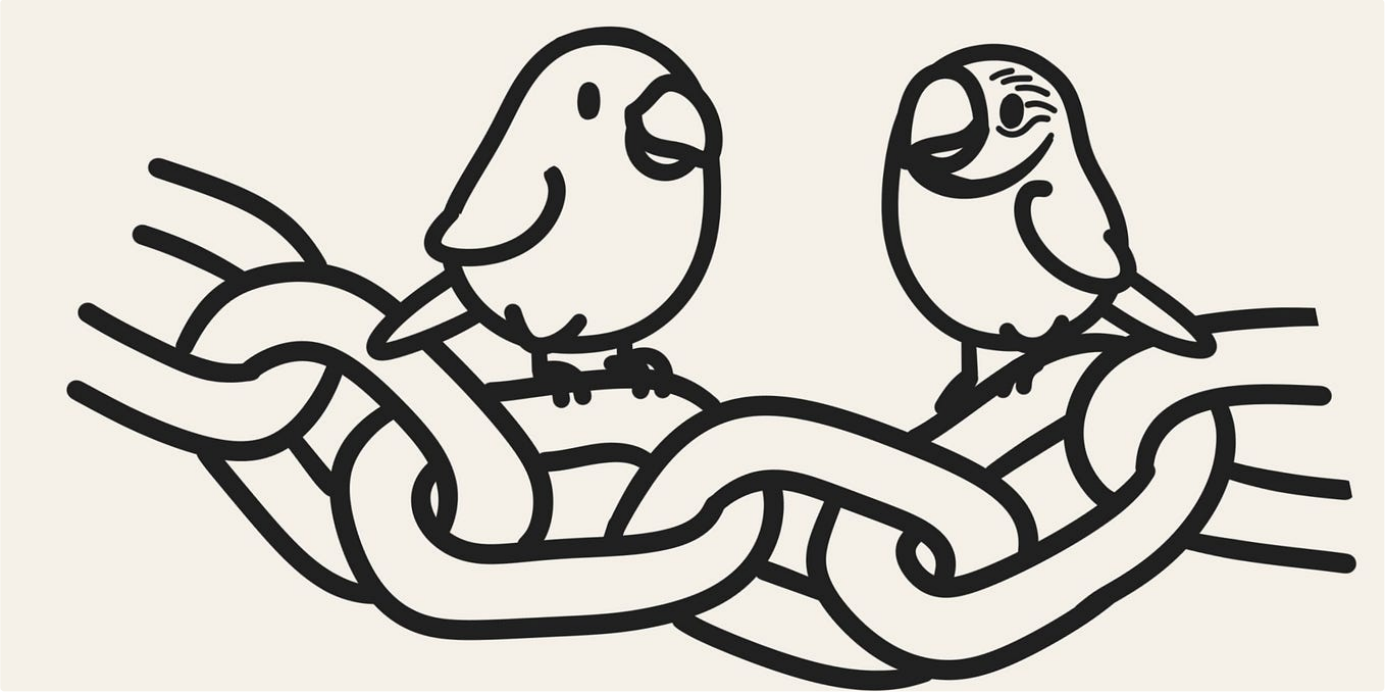
Jacob Marks, Ph.D. in Towards Data Science

How I Turned My Company's Docs into a Searchable Database with OpenAI

And how you can do the same with your docs

15 min read · Apr 25

2.9K 39



 Leonie Monigatti in Towards Data Science

Getting Started with LangChain: A Beginner's Guide to Building LLM-Powered Applications

A LangChain tutorial to build anything with large language models in Python

✨ · 12 min read · Apr 25

 2K  16 

Sai Pavan Yekula

How to Handle Missing Values in your Data

A data scientist hardly ever comes across data with no missing values. Handling such situations is crucial to use the data effectively...

3 min read · Nov 23, 2021

16

See all from Sai Pavan Yekula

See all from Towards Data Science

Recommended from Medium

-	Man	+	Woman	
-	1	+	0	
-	0.3	+	0.3	
-	0.2	+	0.2	
-	0.6	+	0.5	
-	1	+	1	

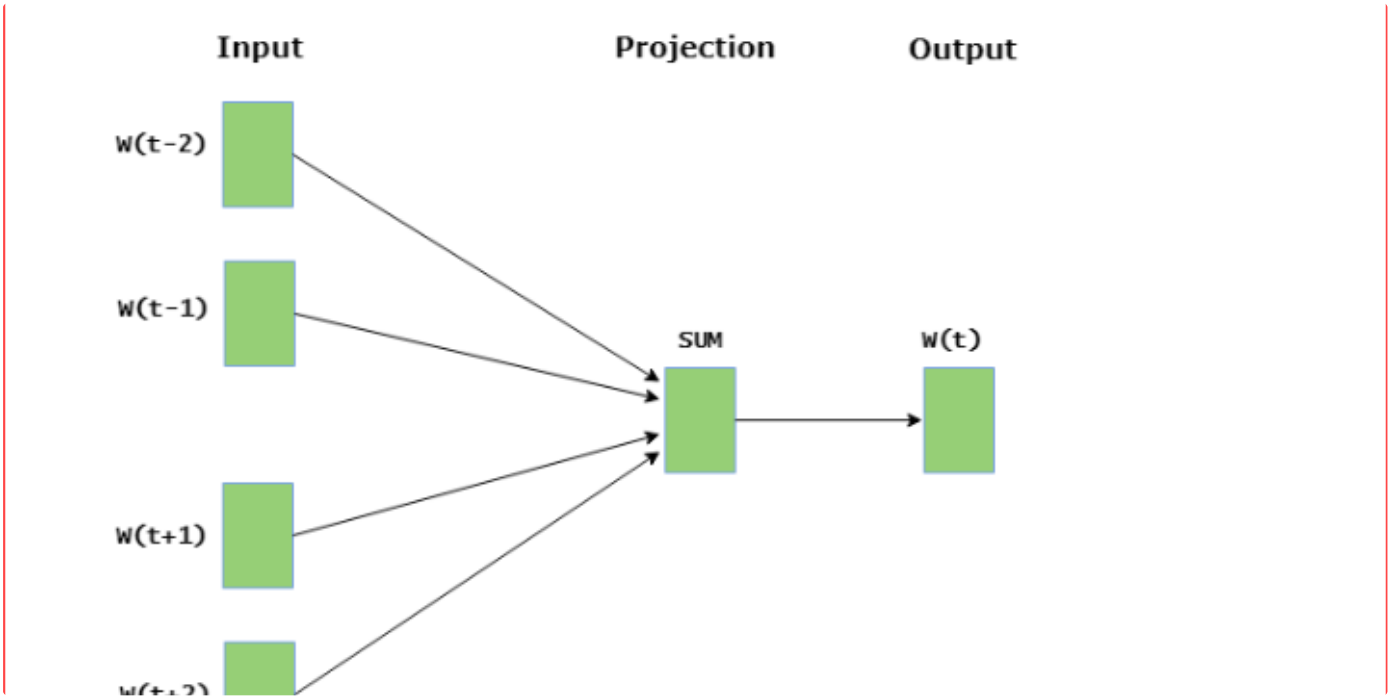
Abhishek Mahli

NLP Word2vec

In Natural Language Processing (NLP), word embedding is a term used for the representation of words for text analysis, typically in the...

3 min read · Jan 16

 114  3



YashwanthReddyGoduguchintha

Build Text Classification Model using Word2Vec(NLP)-Part2

Agenda

10 min read · Dec 5, 2022

11

Lists

What is ChatGPT?

9 stories · 50 saves

Staff Picks

320 stories · 81 saves

Prateek Gaurav

NLP : Zero To Hero [Part 1: Introduction, BOW, TF-IDF & Word2Vec]

Natural Language Processing (NLP) has become an integral part of various industries, including healthcare, finance, and e-commerce, to...

· 10 min read · Mar 23

507 1

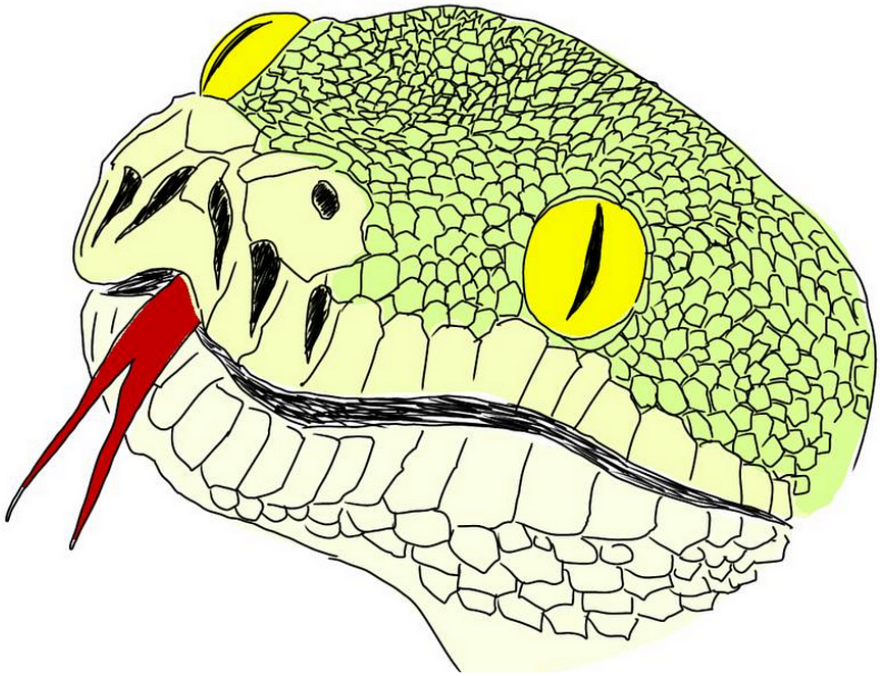
Andrea D'Agostino in Towards Data Science

How to Train a Word2Vec Model from Scratch with Gensim

In this article we will explore Gensim, a very popular Python library for training text-based machine learning models, to train a Word2Vec...

✦ · 9 min read · Feb 6

78



Liu Zuo Lin in Python in Plain English

Python Word2Vec For Text Classification (With LSTM)

Plug & Play Code For Those With No Time

· 5 min read · Feb 4


Chandra Prakash Bathula

Machine Learning Concept 9

Word2Vec , Avg word2vec & TF-IDF word2vec in conversion of word to vector.

2 min read · Feb 21


1