InterviewPrep: Natural Language Processing

Muthu Palaniappan M

December 27, 2023

Agenda

- Text / Feature Representation - 1
  - One Hot Encoding
  - BoW
  - N-Gram
  - TF-IDF
- Text / Feature Representation - 2
  - Word2Vec
    - CBOW
    - Skipgram

Text Representation

- Text representation is a crucial aspect of NLP that involves converting raw text data into machine-readable form.
- Machines are good at numbers. This is a crucial stage where the model generalizations will be a huge Impact.

One-hot Encoding

- Assigns 0 to all elements in a vector except for one, which has a value of 1.
- If i had a sentence, "I love my dog", each word in the sentence would be represented as below:

$$\rightarrow [1\ 0\ 0\ 0], love \rightarrow [0\ 1\ 0\ 0], my \rightarrow [0\ 0\ 1\ 0], dog \rightarrow [0\ 0\ 0\ 1]$$

One-hot Encoding

Advantages

- Easy to Understand.

Disadvantages

- Computationally Expensive
- May not capture the semantic information
- Out of Vocabulary (OOV) Problem

Bag of Words (BoW)

- Each word in the text is considered a feature, and the number of times a particular word appears in the text is used to represent the importance of that word in the text.
- Ignores the grammar and the Word Order.
- Only keep track of the frequency of words.

Bag of Words (BoW)

- The cat in the hat
- The dog in the house
- The Bird in the Sky

Text	dog	cat	bird	in	house	sky	the	hat
The cat in the hat	0	1	0	1	0	0	2	1
The dog in the house	1	0	0	1	1	0	2	0
The bird in the sky	0	0	1	1	0	1	2	0

Figure: Example

Bag of words (BoW)

Advantages

- The length of the vector is fixed - length of the dictionary.

Disadvantages

- Not considering ordering.
- Giving importance to higher order occurring words.

N-Gram

- An N-gram is a traditional text representation technique that involves breaking down the text into contiguous sequences of n-words.
- Bi-gram: Sets of two consecutive words
- Tri-gram: Sets of consecutive 3 words

N-Gram

Example: The dog in the house

- **Uni-Gram**: *"The"*, *"dog"*, *"in"*, *"the"*, *"house"*
- **Bi-Gram**: *"The dog"*, *"dog in"*, *"in the"*, *"the house"*
- **Tri-Gram**: *"The dog in"*, *"dog in the"*, *"in the house"*

N-Gram

Advantages

- Able to capture semantic meaning of the sentence.

Disadvantages

- Computation complexity increases.
- Again OOV is not handled.

TF-IDF

- TF-IDF: Term Frequency-Inverse Document Frequency.
- This is better than BoW since it interprets the importance of a word in a document.
- Weigh words based on how often they appear in a document and how common they are across all documents.

What is Term Frequency? What is IDF?

TF-IDF

$$\textbf{TF-IDF} = \text{Term frequency in document} \times \log\left(\frac{\text{Total number of documents}}{\text{Number of documents containing the term}}\right)$$

Figure: Formulae

TF-IDF

- **Term Frequency**: Number of occurrences of a word in a document divided by a total number of terms in a document.
  - *Example*: You are attending NLP Session. The word NLP occurs: $1/5$
- **Inverse Document Frequency**: Total number of documents in corpus divided by the total number of documents with term T in them and taking the log of a complete fraction.
  - If we have a word that comes in all documents then the resultant output of the log is zero.

TF-IDF

Advantages

- Penalise highly frequent words & low frequency terms in a corpus.

Disadvantages

- Positional information of the word is still not captured in this representation.
- TF-IDF is highly corpus dependent. Cannot handle generalization.

Word Embedding

Representations where contexts and similarities are captured by encoding in a vector space-similar words would have similar representations.

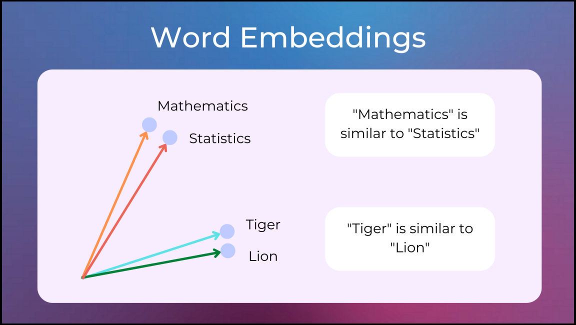

Figure: Word Embedding

Word2Vec

- Deep Learning Technique: used to represent words as continuous vector spaces.
- Words used in similar contexts or having semantic relationships are captured effectively through their closeness in the vector space- effectively speaking similar words will have similar word vectors!
- Created by Google Research Team.

Word2Vec

	battle	horse	king	man	queen	..	woman
authority	0	0.01	1	0.2	1	...	0.2
event	1	0	0	0	0	...	0
has tail?	0	1	0	0	0	...	0
rich	0	0.1	1	0.3	1	...	0.2
gender	0	1	-1	-1	1	...	1

Figure: Feature Representation

Question?: King - Man + Woman = ?

Word2Vec

How it Works?

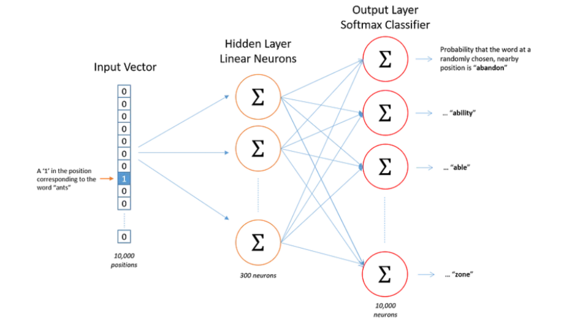

Figure: Two-Layer Depth ANN

- The output layer contains probabilities for a target word given a particular input.
- The hidden weights are treated as the word embedding.

Word2Vec

There are two main architectures for Word2Vec

- Continuous Bag of Words (CBOW)
- Skip-gram.

Word2Vec

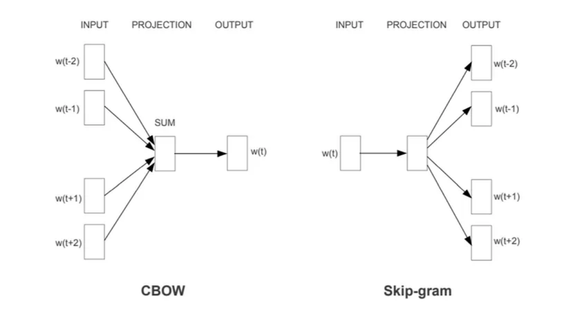

Figure: CBOW vs Skip-gram

Word2Vec - CBOW

Model that predicts a target word based on its context, which is a set of surrounding words.

Architecture

- Input Layer: One-hot encoded vectors representing the context words. Binary Representation - 1 Present in the One-hot encoded.
- Embedding / Hidden Layer: The one-hot encoded vectors are then multiplied by a weight matrix. These weights are trained from the back propagation.
- Output Layer: Softmax layer, which produces a probability distribution over the entire vocabulary. The target word is selected from this distribution.

Word2Vec - Skipgram

Model that predicts the context words (words surrounding a target word) given a target word.

Architecture

- Input Layer: One-hot encoded vectors representing the target words. Binary Representation - 1 Present in the One-hot encoded.
- Embedding / Hidden Layer: The one-hot encoded vectors are then multiplied by a weight matrix. These weights are trained from the back propagation.
- Output Layer: Softmax layer, which produces a probability distribution over the entire vocabulary. The context words is selected from this distribution.

CBOW vs Skipgram

Sentence: *"The cake was chocolate flavoured"*.

- CBOW: "The predict was chocolate flavoured" being inputs and "cake" being the target word.
- Skipgram: Input - "cake" we would expect the model to give us "The", "was", "chocolate", "flavoured" for the given instance.

Note: Skipgrams work well with small datasets and can better represent less frequent words.