Distributed Representations of Words and Phrases and their Compositionality (Negative Sampling)

Authors : Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean

Published in : 2013

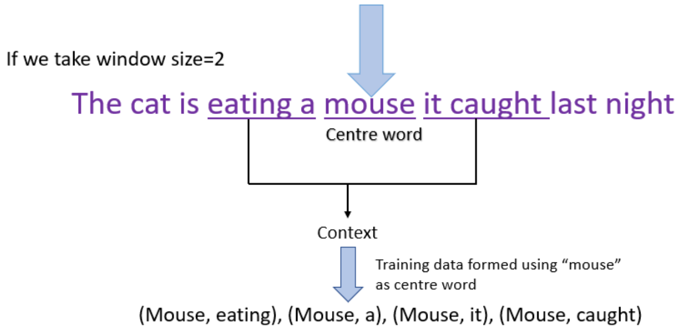
Divya Solanki (MCA , 18)

Vandana Yadav (MCA ,61)

Question 1 : (Total marks = 14)

a) What do you understand by the Skip Gram Model, explain with an example? Explain its architecture with formulations involved. (2+2=4)

- The training objective of the Skip-gram model is **to find word representations** that are useful for **predicting** the **surrounding words** in a sentence or a document.
- The network is going to learn the statistics from the number of times each pairing shows up.



- Architecture of Skip Gram model is :-

For Window Size:- 2

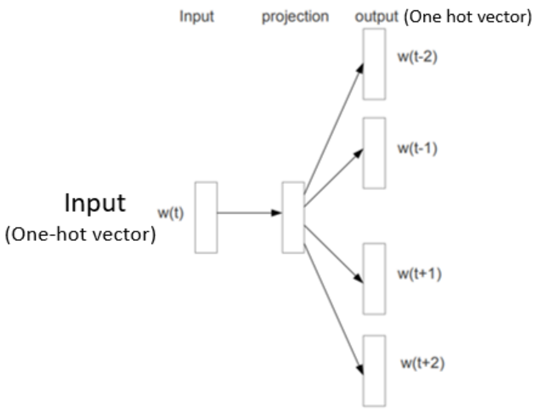


Figure 1: The Skip-gram model architecture. The training objective is to learn word vector representations that are good at predicting the nearby words.

- More formally, given a sequence of training words $w_1, w_2, w_3, \ldots, w_T$,, the objective of the Skip-gram model is to **maximize the average log probability**

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|\underset{\text{Centre word}}{w_t})$$

  - where c is the size of the training context (which can be a function of the center word w_t). Larger c results in more training examples and thus can lead to a higher accuracy, at the expense of the training time.
  - The basic Skip-gram formulation defines $p(w_{t+j}|\ w_t)$ using the softmax function:

Output vector representation

Input vector representation

Given Input Word, what is output word probability?

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^{\top} v_{w_I}\right)}{\sum_{w=1}^{W}\exp\left({v'_w}^{\top} v_{w_I}\right)}$$

Where, v_w and v'_w are the "input" and "output" vector representations of *w*, and W is the number of words in the vocabulary. This formulation is impractical because the cost of computing ∇ log $p(w_O \mid w_I)$ is proportional to W, which is often large (10^{-5} – 10^{-7} terms).

b) What is the major disadvantage of Skip Gram Model, which makes it impractical? (1)

- Summation over the entire vocabulary in the formula of the Skip Gram model is computationally expensive, due to which training time increases.

c) Briefly explain three methods to overcome disadvantage of Skip Gram Model. (3X2=6)

1. <u>**Hierarchical Softmax**</u>

The hierarchical softmax uses a binary tree representation of the output layer with the **W words as its leaves** and, **for each node**, explicitly represents the **relative probabilities of its child nodes**.These define a **random walk** that assigns probabilities to words.

Each word w can be reached by an appropriate path from the root of the tree. Let *n(w, j)* be the j-th node on the path from the root to *w* and *L(w)* be the length of this path, so *n(w, 1)* = root and *n(w, L(w))* = *w*. In addition, for any inner node n, let *ch(n)* be an arbitrary fixed child of *n* and let $[\![x]\!]$ be 1 if *x* is true and -1 otherwise.

$$p(w|w_I) = \prod_{j=1}^{L(w)-1}\sigma\left([\![n(w,j+1) = \mathrm{ch}(n(w,j))]\!]\cdot {v'_{n(w,j)}}^{\top} v_{w_I}\right)$$

where σ(x) = 1/(1 + exp(−x)). It can be verified that $\sum_{w=1}^{W}$ **$p(w \mid w_I)$ = 1**.

This implies that the cost of computing $p(w_O \mid w_I)$ and $\nabla \log p(w_O \mid w_I)$ is proportional to $L(w_O)$, which on average is no greater than $\log W$.

2. Negative Sampling

We define Negative sampling (NEG) by the objective :-

Output vector representation

Input vector representation

Number of negative samples

$$\log \sigma({v'_{w_O}}^\top v_{w_I}) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)}\left[\log \sigma(-{v'_{w_i}}^\top v_{w_I})\right]$$

Sigmoid

This states that wi is taken from a distribution

Word vector of wi

Above equation is used to replace every $\log P(w_O \mid w_I)$ term in the Skip-gram objective. Thus the task is to distinguish the target word w_O from draws from the noise distribution $P_n(w)$ using logistic regression, where there are k negative samples for each data sample.

3. Frequent word subsampling

In very large corpora, the most **frequent words** can easily occur hundreds of millions of times (e.g., “in”, “the”, and “a”).

Such words usually **provide less information value than the rare words**. For example : while the Skip-gram model benefits from observing the co-occurrences of “France” and “Paris”, it benefits much less from observing the frequent co-occurrences of “France” and “the”, as nearly every word co-occurs frequently within a sentence with “the”.
The vector representations of frequent words do not change significantly after training on several million examples.

To counter the imbalance between the rare and frequent words, Use a simple subsampling approach :- each word w_i in the training set is discarded with probability computed by the formula :

Probability to discard word

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$$

t is a chosen threshold (10^{-5})

where $f(w_i)$ is the frequency of word w_i

d) Does structure of the tree used by the Hierarchical softmax has a any effect on performance. Comment. (1)

- The structure of the tree used by the hierarchical softmax has a **considerable effect** on the performance.
- Hierarchical softmax uses Binary Huffman Trees, as they assigns short codes to the frequent words which results in fast training time.

e) How hierarchical softmax is different from standard softmax ? (1)

- Standard softmax evaluates W output nodes in the neural network to obtain the probability distribution, while hierarchical softmax evaluate only about log2(W) nodes.
- Standard softmax assigns two representations v_w (input vector representation) and v'_w (output word representation) to each word *w*, while the hierarchical softmax formulation has one representation v_w for each word *w* and one representation v'_n for every inner node n of the binary tree.

f) What is the main difference between Noise Constructive Estimation and Negative Sampling? (1)

- The main difference between the Negative sampling and NCE is that **NCE** needs **both samples** and the **numerical probabilities** of the **noise distribution**, while **Negative sampling** uses **only samples**.