KALLEL Mahdi
mahdi.kallel@telecom-paris.fr

Lab session # 1
ALTEGRAD 2020

11/23/20

1 Question 1

$$L(t, C_t^+, C_t^-) = \sum_{c \in C_t^+} \log(1 + e^{-w_c . w_t}) + \sum_{c \in C_t^-} \log(1 + e^{w_c . w_t}) \tag{1}$$

$$\frac{\partial L}{\partial w_c} = \begin{cases} \frac{-w_t * e^{-w_c . w_t}}{1 + e^{-w_c . w_t}} \text{ if } c \in C_t^+ \\ \frac{w_t * e^{w_c . w_t}}{1 + e^{-w_c . w_t}} \text{ if } c \in C_t^- \end{cases} = \begin{cases} \frac{-w_t}{1 + e^{w_c . w_t}} \\ \frac{w_t}{1 + e^{-w_c . w_t}} \end{cases}$$

2 Question 2

The gradient of the target word is the sum of the gradients incurred due to classifying both real, and false context words.

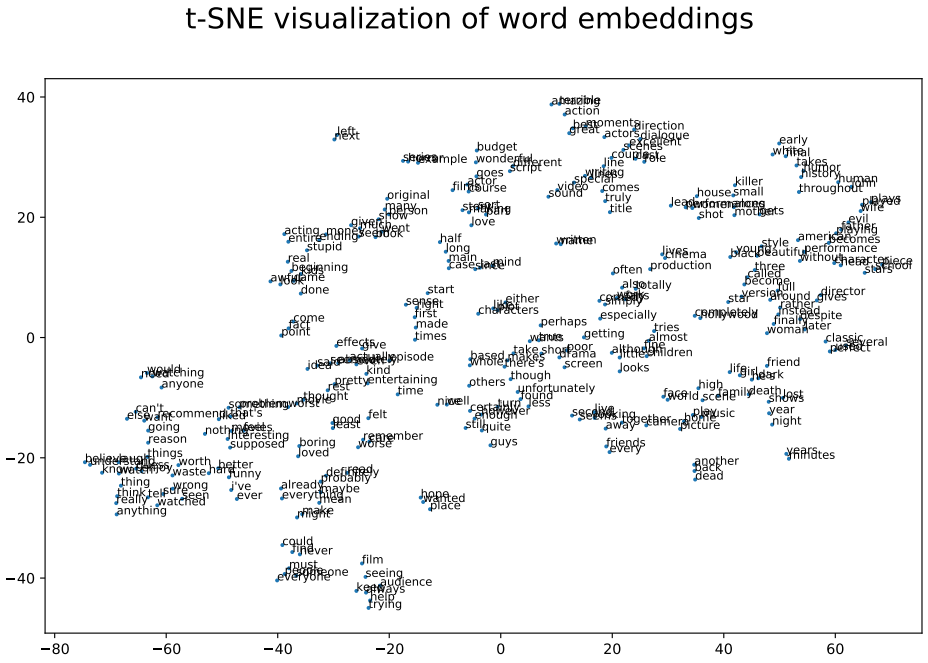
$$\begin{aligned} \frac{\partial L}{\partial w_t} &= \sum_{c \in C_t^+} \frac{\partial}{\partial w_t} \log(1 + e^{-w_c . w_t}) + \sum_{c \in C_t^-} \frac{\partial}{\partial w_t} \log(1 + e^{w_c . w_t}) \\ &= \sum_{c \in C_t^+} \frac{-w_c}{1 + e^{w_c . w_t}} + \sum_{c \in C_t^-} \frac{w_c}{1 + e^{-w_c . w_t}} \end{aligned}$$

3 Question 3

During training, the purpose of the Word2Vec representations is to assign high probabilities to words that appear in the same context.

For example in Fig1 we the words ”Seeing” and ”Film” have close representations as they tend to appear in the same context (Especially when writing movie critiques).

There are also clusters for semantically related words like ”Years” and ”Minutes” . As well as words that would occupy the same syntactic role like ”Instead” / ”Despite”.



To get another point of view, we report some cosine similarity scores in the table below.

The cosine similarity between ”Movie” and ”Film” is 0.99 , meaning these two words have almost identical representations, proving that the model captures synonymity relations. Whereas words that are not ”semantically similar” tend to have very different representations like ”Movie” and ”Banana” we get a score of 0.24.

Word1	Word2	Cosine-Similarity
”Movie”	”Film”	0.99
”Movie”	”Hero”	0.98
”Movie”	”Banana”	0.24
”Boy”	”Girl”	0.98

4 Question 4

Learning the documents representation along with the words representation is done using the ”Distributed Memory Model of Paragraph Vectors (PV-DM)” architecture of [3].

In the reference, they use the Bag of words model, using **many** context words to generate a **single** target which is the following words.

In what follows we propose a slight modification of the (PV-DM) model to use the skip-gram instead of CBOW.

Overall,this does not require any change to the pre-processing pipeline.
On the other hand this would require adding a 3^{rd} matrix W_d each column of this matrix will represent the embedding of a fixed document.

During training, a (target,context) tuple is sampled from each document. The embedding for **this specific document** is concatenated to the **shared** representations of the target word.
Then as in the usual skip-gram model, this new context vector is fed to the ”softmax” layer and we back propagate our error through the network.

During inference time, the word embeddings W_w and the target embedding (softmax) matrix W_t are fixed. We use the same training procedure to back-propagate only through the document embedding matrix W_d untill we reach a stable loss. The resulting embeddings can then be used to the NLP task of choice.