Del Gallo Adriano
delgallo.adriano@gmail.com

Lab session # 2
ALTEGRAD 2019

10/28/19

1 Question 1

Let $(w_c^+, w_c^-) \in C_t^+ \cup C_t^-$
w_c^+ (resp. w_c^-) appears only in one term of the sum over C_t^+ (resp C_t^+) thus :

$$\frac{\partial L}{\partial w_c^+} = \frac{\partial}{\partial w_c^+} \log\left(1 + e^{-w_c^+ . w_t}\right) = -w_t \frac{e^{-w_c^+ . w_t}}{1 + e^{-w_c^+ . w_t}} = -\sigma(-w_c^+ . w_t)\, w_t \qquad \in \mathbb{R}^d$$

Similarly,

$$\frac{\partial L}{\partial w_c^-} = \frac{\partial}{\partial w_c^-} \log\left(1 + e^{w_c^- . w_t}\right) = w_t \frac{e^{w_c^+ . w_t}}{1 + e^{w_c^- . w_t}} = \sigma(w_c^- . w_t)\, w_t \qquad \in \mathbb{R}^d$$

2 Question 2

w_t appears in every term of the sums over C_t^+ and C_t^-.

$$\begin{aligned} \frac{\partial L}{\partial w_t} &= \sum_{c \in C_t^+} -w_c^+ \frac{e^{-w_c^+ . w_t}}{1 + e^{-w_c^+ . w_t}} + \sum_{c \in C_t^-} w_c^- \frac{e^{w_c^- . w_t}}{1 + e^{w_c^- . w_t}} \\ &= \sum_{c \in C_t^+} -w_c^+ \, \sigma(-w_c^+ . w_t) + \sum_{c \in C_t^-} w_c^- \, \sigma(w_c^- . w_t) \qquad \in \mathbb{R}^d \end{aligned}$$

3 Question 3

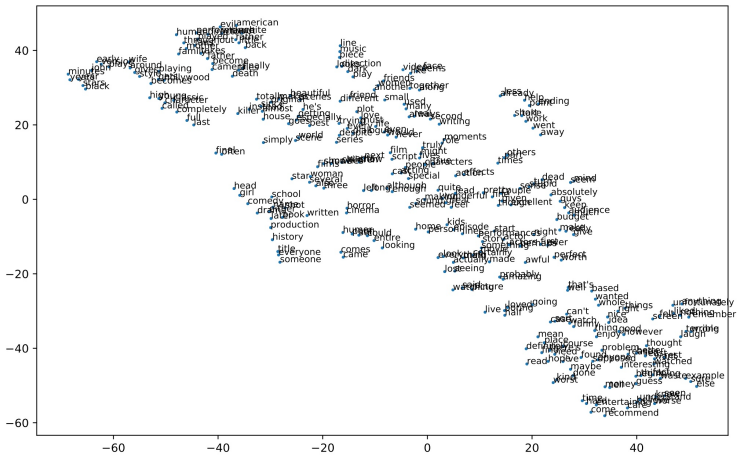

Figure 1: t-SNE representation of embeddings for the 500 most frequent words

Two words being close in the t-SNE representation coincide with having a high cosine similarity. For example for the words 'film' and 'movie', have a similarity of 0.996 and are close in the representation. Low similarity implies a high distance in the embedding space (eg, 'film' and 'banana' with a similarity of 0.439). This comes from the fact that the cosine similarity computes the angle between the two vectors. If they point towards

the same direction in the embedding space, the way they will be used in context is similar and the distance between them will likely be low.
It is also interesting to notice that words with used in similar context tend to form aggregations in the embedding space around a region of common meaning. Moving from one point to another means gradually changing from one meaning to another. The embedding space is continuous in meaning.
We finally see that words that are aggregated sometimes do not have a similar meaning(as in dictionary definitions), which comes from the fact that the training set is very topic specific. With a broader data set with different topics, cosine similarity will truly reflect meaning similarity.

4 Question 4

We wish to modify our pipeline to learn word and paragraph embeddings simultaneously by the skip-gram and negative sampling method. [1] proposes a way to learn paragraph vectors alone in the from a classification problem, where by training the algorithm to select the words correct context given a paragraph. This method is similar to how we learn the word vectors in this lab thus does not require too many changes in our pipeline. We will basically be adding a network running in parallel with a shared output this time predicting the context given an input paragraph. By doing this, we hope that the paragraph vectors will capture the information of the different contexts included in it.
First we must build a corpus from the documents where each index will refer to a unique paragraph. Since we are trying to learn the representation of entire paragraphs and paragraphs are unique in a document (with no way of measuring their relevance beforehand), there is no need for further prepossessing. We note $|C|$ the number of paragraphs in the Corpus.
We have to modify the way we sample training examples to include a reference to the paragraph. Thus a training example is made of (t, p_t, C_t^+, C_t^-), with p_t being the ID of the paragraph in which C_t^+ is contained.
The likelihood has to be changed to take into account the paragraph information with the target word information. This translates into finding :

$$\underset{\theta}{\operatorname{argmax}} \sum_{t=0}^{T} \sum_{c \in C_t^+} log\, p_2(c|t, p_t; \theta)$$

To learn the paragraphs and word vectors independently we take a joint distribution of the form $p_2(c|t, p_t; \theta) = p(c|t; \theta).p^{'}(c|p_t; \theta)$(we don't want the embedding of a paragraph to be dependent on the embedding of the words in it and vice versa) which means we have to introduce another context matrix which will predict context with regards to the paragraph vector in input. We note this matrix W_c' of size $p \times |C|$ with p the embedding dimension which is initialised randomly. We also need a paragraph matrix W_p of size $|C| \times p$ initialised randomly.
The loss function for one training example will be (using negative samlping trick) :

$$L(t, p_t, C_t^+, C_t^-) = \sum_{c \in C_t^+} \log(1 + e^{-w_c^+ . w_t}) + \log(1 + e^{-w_c^{'+} . w_{p_t}}) + \sum_{c \in C_t^-} \log(1 + e^{-w_c^- . w_t}) + \log(1 + e^{w_c^{'-} . w_{p_t}})$$

We have to change our $compute\_dot\_products$, $compute\_gradients$ and $compute\_loss$ accordingly and run the training. The paragraph embeddings will be available in W_p.

References

[1] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *In International conference on machine learning*, page 1188–1196, 2014.