Khaoula Belahsen
khaoula.belahsen@yahoo.fr

Lab session # 3
ALTEGRAD 2019

11/04/19

1 Question 1

The log loss is defined as :

$$-\frac{1}{N}\sum_{i=1}^{N}[y_i \log p_i + (1-y_i)\log(1-p_i)]. \tag{1}$$

We code and compute the logloss function for the 3 different states.

We use a prediction of 0.9 for confident correct prediction, 0.5 for the unsure correct and 0.1 for rhe strongly incorrect.

LogLoss(1,0.5)
0.69315 → medium penalization

LogLoss(1,0.9)
0.10536 → very small penalization

LogLoss(1,0.1)
2.3026 → big penalization

2 Question 2

The missing value is -3.
It is obtained by calculating the convolution between our input and the 1^{st} filter. the missing value corresponds to the second term of the output of the first filter. Which is the result of the following computation :

$$\begin{pmatrix}0 & 1\\ 2 & 1\\ 2 & 2\end{pmatrix} \text{ *term to term * } \begin{pmatrix}0 & 0\\ -1 & 0\\ -1 & 0\end{pmatrix} = \begin{pmatrix}0 & 0\\ -2 & 0\\ -2 & 0\end{pmatrix} \rightarrow sum = -4 + bias = -3$$

3 Question 3

We can use a sigmoid with 1 unit.
We can use also use reLU or a tan2h

4 Question 4

The number of parameters to train can be divided into :

- For the input, we have sxd parameters.
- For each filter : hxd parameters so $n_f * \sum h_i * d$ parameters in total
- $n_f * 1$ baises, one for each fitler.
- Output*input dimension for the activation function
- 1 for the final bias

The final formula for the number of trainable parameters is :

$$s * d + n_f * \sum h_i * d + n_f * 1 + output_{activ} * input_{activ} + 1$$

5 Question 5

Before training our model, the t-SNE visualization shows no difference of embeddings between the 0 and 1 classes. The resulting plots are scattered all across the space with no visible distinction. We train our model over 2 epochs to avoid over-fitting and obtain a final accurcy of 0.7604.

From the figure of the after training plot, we see a distinct separation between the 0 and 1 classes. The 0 class is located a the right and the 1 class to the left. The model is able to capture and encode the information contained in the labels making the two classes separable.

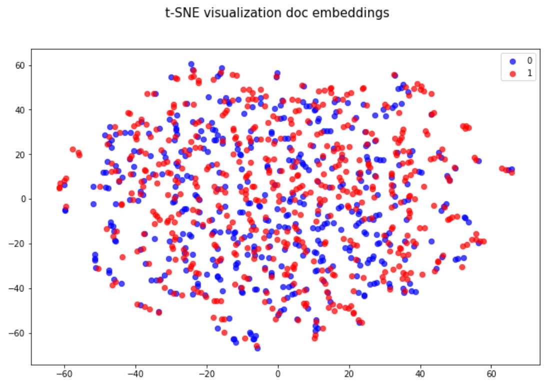

Figure 2: Before training.

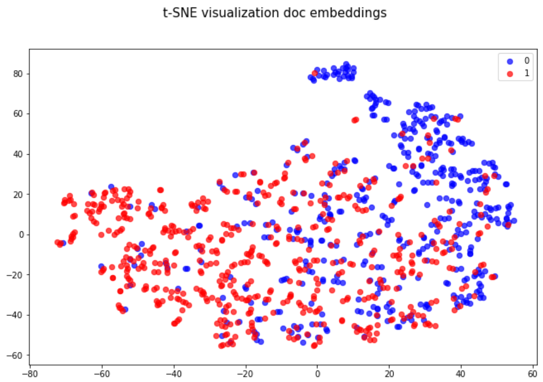

Figure 3: Before training.

6 Question 6

By looking at the saliency map, we observe that the instances ”disappointment” and ”worst movie ever” identify as been the most important in setting the meaning of the class. Following the same logic, ”oh god”, ”this was” , ”bucks” and others are the least important in preserving the whole meaning.

The review text was : ”Oh , god , this was such a disappointment ! Worst movie ever . Not worth the 15 bucks .” It is clear that it corresponds to a bad review of the movie and the most important words to understand the meaning are indeed : ”disappointment” and ”Worst movie ever”.

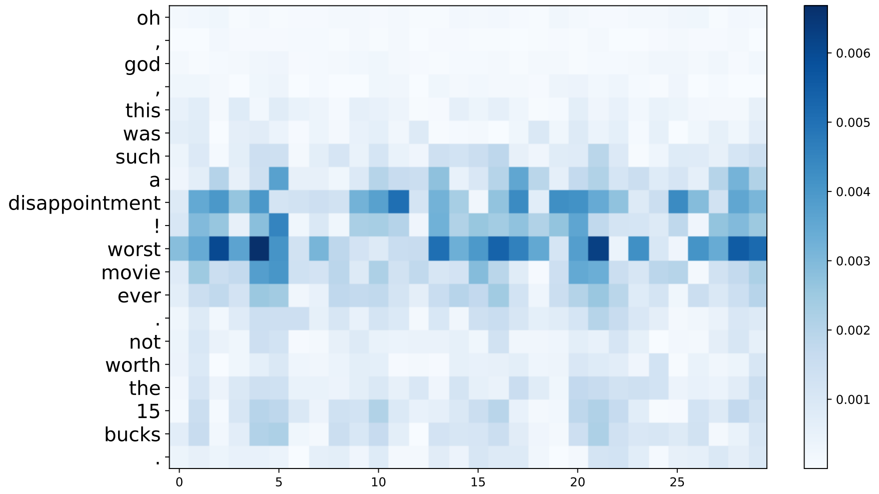

Figure 4: Saliency map.

7 Question 7

The CNN model clearly does a good job at extraction features from word and document embeddings. However, it has some clear limitations that are specific to natural language processing.

Indeed, in text and document procesing, we care about the positiong of a word in a sentence and its proximity with other words in the sentence. Unfortunately, the CNN does not keep the spatial position of the words after the pooling step. It struggle to preserve sequential order.

One other limitation or weakness about CNNs is their tendency to overfit with smaller datasets. Indeed, CNNs have a lot of parameters to train as was shown in **Question 3**, and with smaller datasets, they will struggle to perform a non overfitted prediction.