VU Thi Hai Yen
haiyen96.hp@gmail.com

Lab session # 3
ALTEGRAD 2019

11/03/19

1 Question 1

For an observation with true label $y = 1$, a probability prediction p to class 1 gives the log loss:

$$\text{logloss} = -y \log(p) - (1 - y) \log(1 - p) = -\log(p) \tag{1}$$

- For a correct prediction, $p \approx 1$, $\text{logloss} \approx -\log(1) = 0$.
- For an unsure correct prediction, $p \approx 0.5$, $\text{logloss} \approx -\log(0.5) \approx 0.69$.
- For a strongly incorrect prediction, $p \approx 0$, $\text{logloss} \approx -\log(0) = +\infty$. In fact, for a relatively small $p \approx 0.0001$, we have already $\text{logloss} \approx 9.21$.

2 Question 2

We have the input matrix $\mathbf{A} \in \mathbb{R}^{7\times 2}$, the filter $\mathbf{W}_0 \in \mathbb{R}^{3\times 2}$ and its bias b_0 is defined as follows:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 2 & 1 \\ 2 & 2 \\ 2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{W}_0 = \begin{bmatrix} 0 & 0 \\ -1 & 0 \\ -1 & 0 \end{bmatrix}, b_0 = 1. \tag{2}$$

Then, the output of this filter is a vector $\mathbf{o}^{(0)} \in \mathbb{R}^5$ where:

$$\mathbf{o}_i^{(0)} = \mathbf{W}_0 \cdot \mathbf{A}[i : i + 2, :] + b_0. \tag{3}$$

The missing value in the Figure 1 is therefore:

$$\mathbf{o}_2^{(0)} = \begin{bmatrix} 0 & 0 \\ -1 & 0 \\ -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 \\ 2 & 1 \\ 2 & 2 \end{bmatrix} + 1 = -3. \tag{4}$$

3 Question 3

As we are performing a binary classification task, we can use the sigmoid activation function at the final layer. The output of sigmoid is the probability that the label is equal to 1 (or -1):

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

In this case, we need only one unit in the output layer instead of two as if we use softmax. This helps to reduce $all\_n_f + 1$ parameters in the model, where all_n_f is the total number of filters in all the branches.

4 Question 4

We sketch the model as follows:

$$\begin{aligned} input(s \times 1) \xrightarrow{\textbf{emb\_matrix(V+2}\times\textbf{d)}} v\_emb(s \times d) \xrightarrow[n_f \textbf{ biases (1)}]{n_f \textbf{ filters (h}\times\textbf{d)}} n_f feature\_maps(s - h + 1 \times 1) \\ \xrightarrow[\textbf{Concat (0)}]{\textbf{1D MaxPool (0)}} v\_feature(n_f \times 1) \xrightarrow{\textbf{Dense (2n}_\textbf{f}\textbf{+2 if softmax, n}_\textbf{f}\textbf{+1 if sigmoid)}} output. \end{aligned} \tag{6}$$

The parameters of the model are shown on the arrows and are in **bold**. It is then easy to see that the number of parameters of the model is:

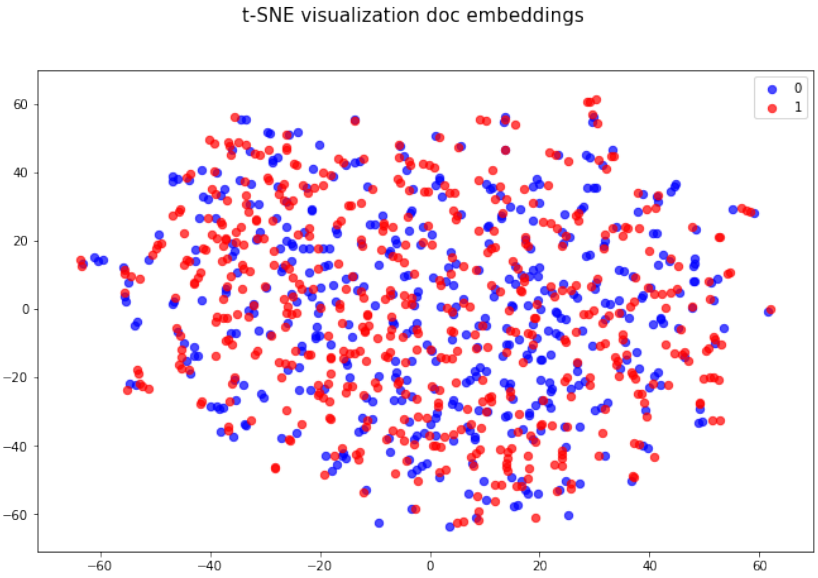
- $(V+2)d + n_f(hd+1) + 2n_f + 2$ if using softmax activation for output.
- $(V+2)d + n_f(hd+1) + n_f + 1$ if using sigmoid activation for output.

If we suppose further that we use $n_{f1}, n_{f2}, \ldots, n_{fk}$ filters with size $h_1, h_2, \ldots, h_n$ respectively. The total number of parameters is:

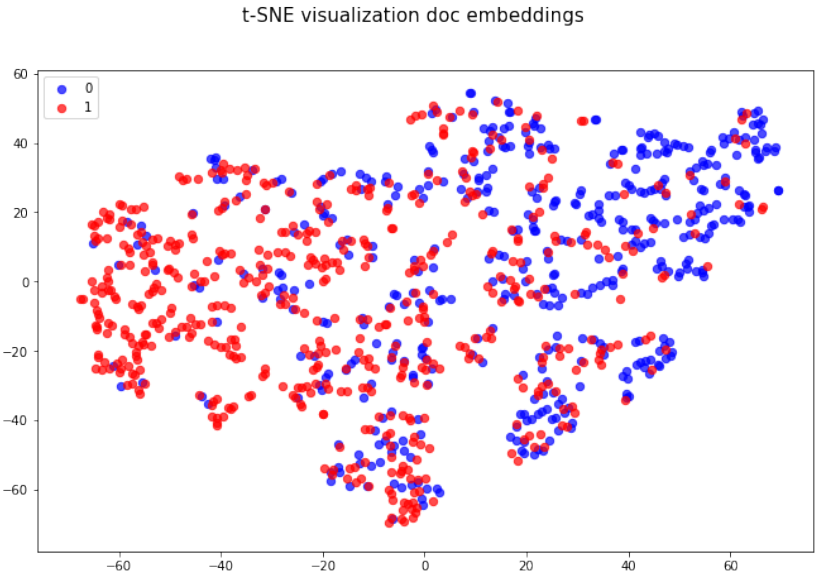
- $(V+2)d + \sum_{i=1}^{k} n_{fi}(h_i d + 1) + 2(\sum_{i=1}^{k} n_{fi}) + 2$ if using softmax activation for output.
- $(V+2)d + \sum_{i=1}^{k} n_{fi}(h_i d + 1) + \sum_{i=1}^{k} n_{fi} + 1$ if using sigmoid activation for output.

5 Question 5

We show the document embeddings before training:



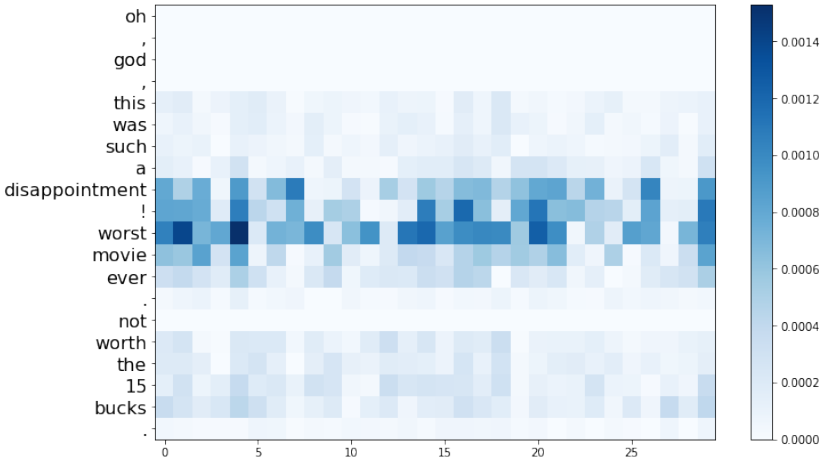
We see that the embedded document vectors are mixed together as the embedding matrix is randomly initialized at the beginning of the training. After training, we obtain the following embeddings:



We can see that there is a separation between the two classes of documents. The documents with label 1 are on the left while documents with label 0 are generally on the right.

6 Question 6

We obtain the following saliency maps:



We see a huge value of saliency at the word **worst**, which is reasonable because such word usually appears at the negative comments and therefore can mostly defines the label of the documents. Another word with high magnitudes is **disappointment**, which also often shows up in negative comments. The irrelevant words almost does not affect the label of the documents and therefore have very low magnitudes. Note that as we use filter size of 3 and 4, some neighbor words of the striking words also have some high values.

7 Question 7

We list some limitations of the CNN model:

- As we only use fixed-size filters, some long-range dependency between words may not be captured.
- The feature maps obtained after sliding the filters through the input are then passed through a MaxPooling, which could loss sequential information of the words in the document.
- May require many filters with different size to capture all the behaviors of the documents, which could lead to a lot of parameters to be trained.
- However, using many filters could lead to the problem of overfitting.