Ioana-Sabina STOIAN
sabina.stoian@gmail.com

Lab session # 3
ALTEGRAD 2019

11/04/19

1 Question 1

For a given observation ($N = 1$) with $y_i = 1$, the loss equation is equal to:

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log_2(p_i) + (1 - y_i)\log_2(1 - p_i)) =$$
$$= -\frac{1}{1}\sum_{i=1}^{1}\log_2(p_i) = -\log_2(p_i)$$

- Confident correct prediction $p_i = 0.95$

$$logloss = -\log_2(0.95) \approx 0.074$$

- Unsure correct prediction $p_i = 0.51$

$$logloss = -\log_2(0.51) \approx 0.971$$

- Strongly incorrect prediction $p_i = 0.1$

$$logloss = -\log_2(0.1) \approx 3.321$$

We can observe from the results that: if we have a confident correct prediction, the loss for that example is very low (≈ 0.074) because the model has predicted properly enough, if we have an unsure correct prediction, the loss has an intermediate value of ≈ 0.971 and if we have strongly incorrect prediction it means that the model performed poorly on this example and the associated loss will have a higher value (which, in our case, is ≈ 3.321). The three results for the loss are very intuitive, because we would expect to have a higher loss when the model performs poorly, with the loss decreasing as the model's prediction gets better.

2 Question 2

The missing value can be computed as described in the following. Because we need to compute a value from the first feature map, this means that the result will be the convolution of the input with the first filter (W_0). Because the unknown value is the second value from the feature map, this means that we need to perform convolution on the inputs from position 1 to position 1+3 = 4 (3 is the size of the filter). Because the input has two channels, we need to compute the convolution for every channel (channel-wise), the results will then be summed up and we will add the bias (b_0). The operation $\odot$ is the Hadamard product, which computes the element-wise multiplication between the two vectors. The operation sum sums all the elements from the resulting vector (after the element-wise multiplication).

$$sum(\begin{bmatrix}0\\2\\2\end{bmatrix} \odot \begin{bmatrix}0\\-1\\-1\end{bmatrix}) + sum(\begin{bmatrix}1\\1\\2\end{bmatrix} \odot \begin{bmatrix}0\\0\\0\end{bmatrix}) + 1 = sum(\begin{bmatrix}0\\-2\\-2\end{bmatrix}) + sum(\begin{bmatrix}0\\0\\0\end{bmatrix}) + 1 = (0+(-2)+(-2)) + (0+0+0) + 1 = -3$$

So the missing value is equal to -3.

3 Question 3

In order to perform binary classification, we can choose as the activation for the last layer the softmax function (as presented in the document) or the logistic (sigmoid) function, $\sigma(x) = \frac{1}{1+e^{-x}}$. By using the softmax function, the number of units in the final layer will be equal to 2 (the softmax would output 2 values, which are normalized and represent the probability of each class). If we use the sigmoid function, we will have one unit as output (its output p can be interpreted as the probability of class 1, with $1 - p$ being the probability of class 0).

4 Question 4

$n_f =$ number of filters per branch
$n_b = 1$, number of branches
$d =$ dimensionality of word embeddings
$h =$ filter sizes
$V =$ size of the vocabulary
Max-pooling $\rightarrow 0$ trainable parameters
Dropout $\rightarrow 0$ trainable parameters
Number of trainable parameters of the embedding layer $= d \cdot V$ (if we consider that the padding and OOV labels are included in the dictionary; otherwise, the number of trainable parameters would be $d \cdot (V + 2)$).
Number of trainable parameters from the convolution layer $= n_b \cdot n_f \cdot h \cdot d + n_b \cdot n_f$
Affine layer with softmax $\rightarrow n_b \cdot n_f \cdot 2 + 2$ trainable parameters
Total number of parameters $= d \cdot V + n_f \cdot h \cdot d + n_f + n_f \cdot 2 + 2$ (if we consider $n_b = 1$ and that the padding and OOV labels are included in the dictionary).

5 Question 5

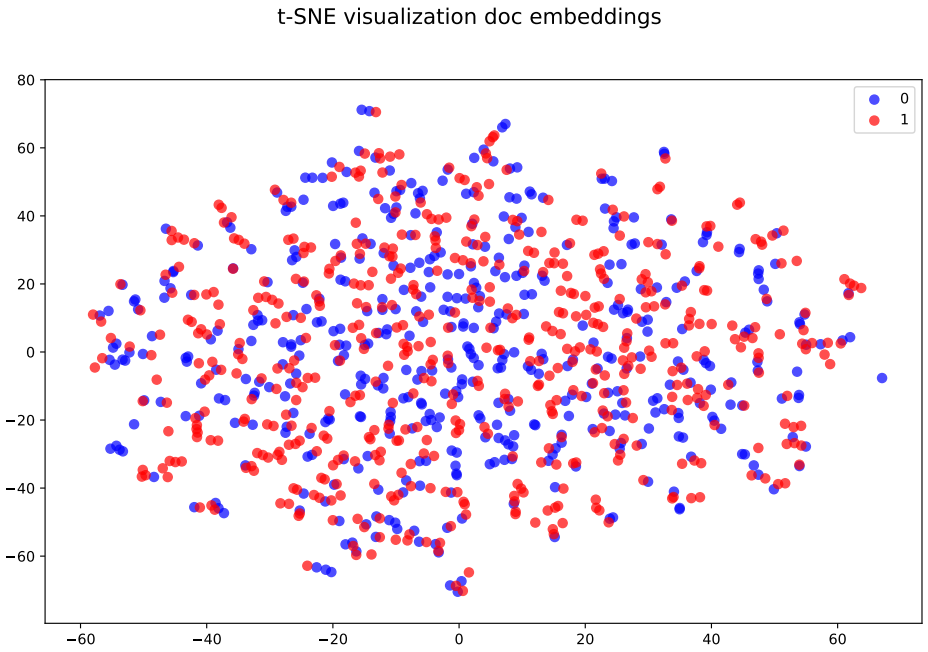

Figure 1: t-SNE visualization of the document embeddings before training.

In Figure 1, we have presented the t-SNE visualization of the document embeddings before training. It can be seen from the plot that it's not easy to distinguish any visual structure (e.g. no clustering), thus it is hard to categorize a point from the plot based on its position. Because the network is initialized randomly, the document embeddings can not describe the semantics of the document. Therefore, the points from both classes are spread all over the plot because they do not have distinctive features that can allow for a proper classification.
In Figure 2, we have presented the t-SNE visualization of the document embeddings after 2 epochs of training. It is easily observable that the document representations are more clustered and the documents belonging to the same class appear more grouped in this plot. Because the network is trained to perform sentiment detection, it captures semantic details from the documents. Thus, the documents that provide similar sentiment information will appear closer to one another in the plot. This information is necessary and useful for a proper classification of a document into one of the two classes (positive or negative review).

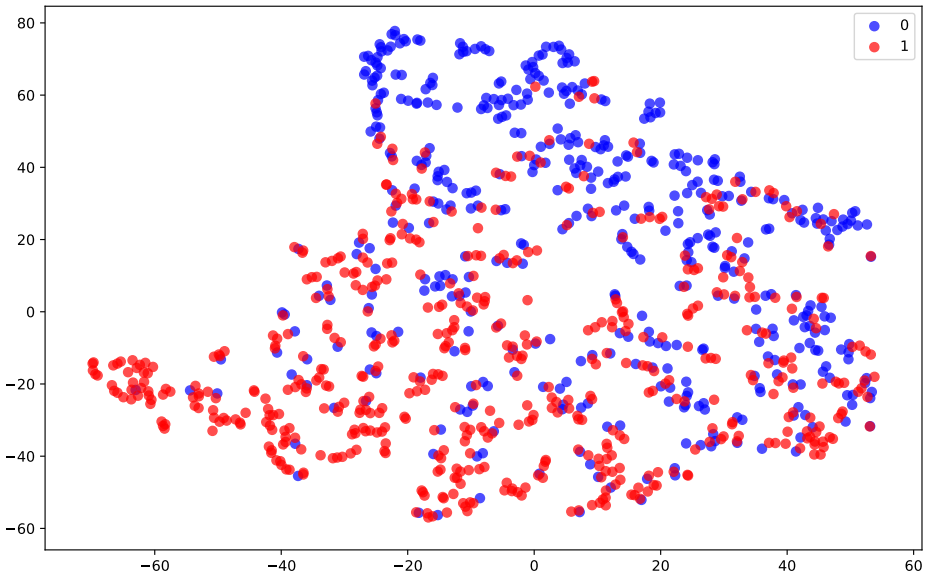

Figure 2: t-SNE visualization of the document embeddings after training for 2 epochs.

6 Question 6

In Figure 3, we present the saliency map for a test review. For every word from the review, we have computed the partial derivative of the model's prediction with respect to the word's embedding and for every component of the resulting gradient we have computed its absolute value (magnitude). Thus, every row of the saliency map presents the magnitude of the gradient corresponding to one word embedding and every component of the gradient is colored based on the magnitude of the gradient. Components with higher magnitudes are attributed darker colors and components with lower magnitudes are attributed lighter colors. A higher magnitude intuitively means that that specific component of the embedding has a higher influence in computing the prediction of the model because if it were changed a little, it would induce considerable changes in the prediction. We can observe that the word associated with the highest number of high magnitude components is the word "worst" and this follows our intuition, because the test review is negative and the word "worst" suggests negative sentiment. It is very plausible that this word will appear in a negative review, so the prediction of the model is influenced a lot by this word and, thus, it is very relevant for the classification of the document. Words like "the" or "oh" have almost all components painted in lighter colors, so this means that they are not relevant for the prediction of the class. This fact is also very intuitive, because words like these two can appear in both positive and negative reviews, which means that they will not weigh considerably when predicting the class.

7 Question 7

This CNN model has the following disadvantages. First, the position of the words in the phrase is lost after a max-pooling layer (the max-pooling layer selects and preserves the maximum value, but does not memorize its position). However, the order of words does matter in a sentence because it is relevant for capturing the semantics of the sentence (e.g. the meaning of a review does depend on its word order). Second, the size of the receptive field of a CNN is also an important criterion when trying to capture the semantics of some text. If the receptive field is small, this means that the correlations between some words (which may not be close enough to one another in a sentence or a review) will not be sufficiently exploited, even if sometimes it can be really helpful when we try capture the semantics of some text. By better connecting more words, we can potentially better capture some subtle details, like sarcasm and irony, that can be important for the classification task.

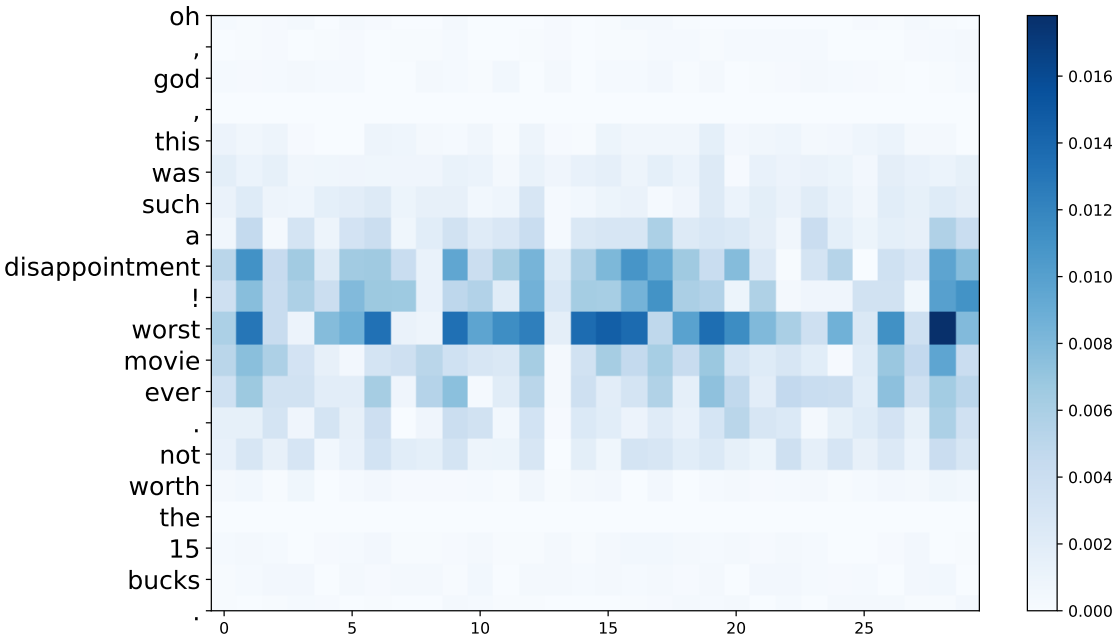

Figure 3: Saliency map