Natural Language Processing, Assignment 1

September 26th, 2023

Problem 1 : (56 points) Let's have a quick refresher on the `word2vec` algorithm. The key insight behind `word2vec` is that *'a word is known by the company it keeps'*. Concretely, suppose we have a 'center' word c and a contextual window surrounding c. We shall refer to words that lie in this contextual window as 'outside words'. For example, in Figure 1 we see that the center word c is 'banking'. Since the context window size is 2, the outside words are 'turning', 'into', 'crises', and 'as'.

The goal of the skip-gram `word2vec` algorithm is to accurately learn the probability distribution $P(O|C)$. Given a specific word o and a specific word c, we want to calculate $P(O = o|C = c)$, which is the probability that word o is an 'outside' word for c, i.e., the probability that o falls within the contextual window of c.

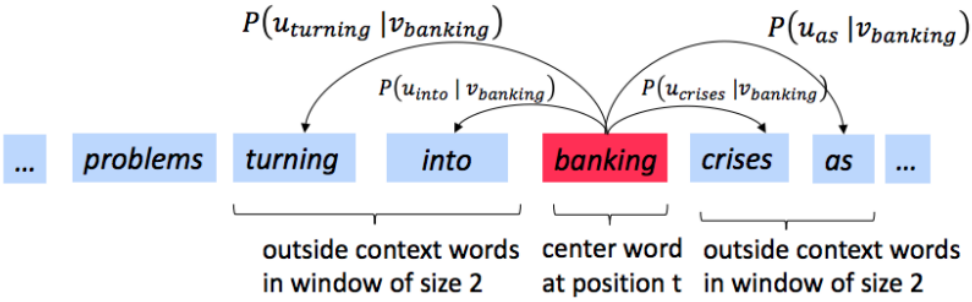

Figure 1: The word2vec skip-gram prediction model with window size 2

In `word2vec`, the conditional probability distribution is given by taking vector dot-products and applying the softmax function:

$$P(O = o|C = c) = \frac{\exp(\boldsymbol{u}_o^T \boldsymbol{v}_c)}{\sum_{w \in \text{Vocab}} \exp(\boldsymbol{u}_w^T \boldsymbol{v}_c)} \tag{1}$$

Here, $\boldsymbol{u}_o$ is the 'outside' vector representing outside word o, and $\boldsymbol{v}_c$ is the 'center' vector representing center word c. To contain these parameters, we have two matrices, $\boldsymbol{U}$ and $\boldsymbol{V}$. The columns of $\boldsymbol{U}$ are all the 'outside' vectors $\boldsymbol{u}_w$. The columns of $\boldsymbol{V}$ are all of the 'center' vectors $\boldsymbol{v}_w$. Both $\boldsymbol{U}$ and $\boldsymbol{V}$ contain a vector for every $w \in$ Vocabulary.[1]

Recall from lectures that, for a single pair of words c and o, the loss is given by:

$$J_{\text{naive-softmax}}(\boldsymbol{v}_c, o, \boldsymbol{U}) = -\log P(O = o|C = c) \tag{2}$$

We can view this loss as the cross-entropy[2] between the true distribution $\boldsymbol{y}$ and the predicted distribution $\hat{\boldsymbol{y}}$. Here, both $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ are vectors with length equal to the number of words in the vocabulary. Furthermore, the k^{th} entry in these vectors indicates the conditional probability of the k^{th} word being an 'outside word'

[1] Assume that every word in our vocabulary is matched to an integer number k. Bolded lowercase letters represent vectors. $\boldsymbol{u_k}$ is both the k^{th} column of $\boldsymbol{U}$ and the 'outside' word vector for the word indexed by k. $\boldsymbol{v_k}$ is both the k^{th} column of $\boldsymbol{V}$ and the 'center' word vector for the word indexed by k. **In order to simplify notation we shall interchangeably use k to refer to the word and the index-of-the-word.**

[2] The Cross Entropy Loss between the true (discrete) probability distribution p and another distribution q is $-\sum_i p_i \log(q_i)$.

for the given c. The true empirical distribution $\boldsymbol{y}$ is a one-hot vector with a 1 for the true outside word o, and 0 everywhere else. The predicted distribution $\hat{\boldsymbol{y}}$ is the probability distribution $P(O|C = c)$ given by our model in equation (1).

1. (10 points) Show that the naive-softmax loss given in Equation (2) is the same as the cross-entropy loss between $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$; i.e., show that

$$-\sum_{w \in \text{Vocab}} y_w \log(\hat{y}_w) = -\log(\hat{y}_o). \tag{3}$$

   Your answer should be one line.

2. (16 points) Compute the partial derivative of $J_{\text{naive-softmax}}(\boldsymbol{v}_c, o, \boldsymbol{U})$ with respect to $\boldsymbol{v}_c$. Please write your answer in terms of $\boldsymbol{y}$, $\hat{\boldsymbol{y}}$, and $\boldsymbol{U}$. Note that in this course, we expect your final answers to follow the shape convention.[3] This means that the partial derivative of any function $f(x)$ with respect to x should have the same shape as x. For this subpart, please present your answer in vectorized form. In particular, you may not refer to specific elements of $\boldsymbol{y}$, $\hat{\boldsymbol{y}}$, and $\boldsymbol{U}$ in your final answer (such as $\boldsymbol{y_1}$, $\boldsymbol{y_2}$,...).

3. (16 points) Compute the partial derivatives of $J_{\text{naive-softmax}}(\boldsymbol{v}_c, o, \boldsymbol{U})$ with respect to each of the 'outside' word vectors, $\boldsymbol{u}_w$'s. There will be two cases: when $w = o$, the true 'outside' word vector, and $w \neq o$, for all other words. Please write your answer in terms of $\boldsymbol{y}$, $\hat{\boldsymbol{y}}$, and $\boldsymbol{v}_c$. In this subpart, you may use specific elements within these terms as well, such as $(\boldsymbol{y_1}, \boldsymbol{y_2}, ...)$.

4. (4 points) Compute the partial derivative of $J_{\text{naive-softmax}}(\boldsymbol{v}_c, o, \boldsymbol{U})$ with respect to $\boldsymbol{U}$. Please write your answer in terms of $\frac{\partial J(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{u_1}}$, $\frac{\partial J(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{u_2}}$, ..., $\frac{\partial (\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{u}_{|\text{Vocab}|}}$. The solution should be one or two lines long.

5. (10 points) Now we shall consider the Negative Sampling loss, which is an alternative to the Naive Softmax loss. Assume that K negative samples (words) are drawn from the vocabulary. For simplicity of notation we shall refer to them as $w_1, w_2, ..., w_K$ and their outside vectors as $\boldsymbol{u}_1, ..., \boldsymbol{u}_K$. For this question, assume that the K negative samples are distinct. In other words, $i \neq j$ implies $w_i \neq w_j$ for $i, j \in \{1, ..., K\}$. Note that $o \notin \{w_1, ..., w_K\}$. For a center word c and an outside word o, the negative sampling loss function is given by:

$$J_{\text{neg-sample}}(\boldsymbol{v}_c, o, \boldsymbol{U}) = -\log(\sigma(\boldsymbol{u}_o^T \boldsymbol{v}_c)) - \sum_{k=1}^{K} \log(\sigma(-\boldsymbol{u}_k^T \boldsymbol{v}_c)) \tag{4}$$

   for a sample $w_1, ..., w_K$, where $\sigma(\cdot)$ is the sigmoid function.[4]

   Please repeat parts (2) and (3), computing the partial derivatives of Jneg-sample with respect to $\boldsymbol{v}_c$, with respect to $\boldsymbol{u}_o$, and with respect to a negative sample $\boldsymbol{u}_k$. Please write your answers in terms of the vectors $\boldsymbol{u}_o$, $\boldsymbol{v}_c$, and $\boldsymbol{u}_k$, where $k \in [1, K]$. After you've done this, describe with one sentence why this loss function is much more efficient to compute than the naive-softmax loss.

Problem 2 : (12 points) Given an input sentence, the problem of automatic spelling correction aims to output a corrected version. Formulate this problem in the source-channel paradigm:

1) Draw the source-channel diagram.
2) Write out the Bayes classifier, just like we did in class.
3) Explain the different components.

[3] This allows us to efficiently minimize a function using gradient descent without worrying about reshaping or dimension mismatching. While following the shape convention, we're guaranteed that $\theta := \theta - \alpha \frac{\partial J(\theta)}{\partial \theta}$ is a well-defined update rule.

[4] Note: the loss function here is the negative of what Mikolov et al. had in their original paper, because we are doing a minimization instead of maximization in our assignment code. Ultimately, this is the same objective function.

Problem 3 : (32 points) In class we discussed the heterogeneity of natural language with regard to register, time period, dialect, domain, topic, discourse act, speaker, audience, etc. Each particular combination of the above attributes yields a different "sub-language".

Locate training material for two (or more) such sub-languages (at least 100K words each, preferably more like 1M), and write a simple program to collect simple statistics, such as word counts, sentence lengths, word combinations, and perhaps letter distributions. The goal is to think of statistical differences between the two sub-languages and collect textual statistics sufficient to test for those differences. For example, you may choose to measure the richness of the vocabulary (as measured by how many different words are present in a fixed amount of writing) in the two sub-languages and compare them. Furthermore, you may look at any other measurable aspects of the text that differ between the two sets, such as the distribution of word lengths, the 20 most frequently used words in each set, or (more interestingly) words that frequently appear in one set and appear infrequently in the other.

Try to use sub-languages where you suspect interesting differences, and make two hypotheses:

- one about the relationship between the richness of vocabulary used in each sub-language (such as which uses a larger vocabulary or contains more rare/uncommon words), and
- one about anything else that you suspect may be interesting.

Finally, using the statistics computed for each sub-language, confirm or refute your hypotheses.

You are encouraged to collect training material from the Internet or other sources of text "in the wild". Alternatively, you may use a corpus with which you are familiar. However, please try not to use the same data sets as other students. Of particular interest are sub-languages that differ along a single attribute. Languages other than English are fine, too. (Italian, Sindarin and Dothraki are all acceptable.)

Hand in:

1. Descriptions of your sub-languages, including their word-counts;
2. Descriptions of any preprocessing of the data, if applicable;
3. All computed statistics for the two sub-languages; and
4. Your two hypotheses, and whether they were confirmed or refuted.

You do not need to submit your code or corpora. But if possible, please include a pointer to your corpora in your submission. Interesting submissions may be used as examples in future years (with your permission and credited to you).