NLP Course 097215 - Winter 2022-23 - HW2-Dry

1 Word2Vec

In this assignment we derive the famous Word2Vec model [MSC+13]. This model is simply a language model, but here we shall describe it as follows:

- Every word w is associated with two vectors $w_o, w_c \in R^d$. The coordinates of these vectors are the parameters of the model.
- The probabilities of the language model are given as a function of the vector representations of the words in the corpus.
- And, finally, our goal is to estimate the model parameters (vector representations for the participating words) that optimize the language model objective. Optimization is done with stochastic gradient descent (gradient descent where optimization is done over one example in each iteration).

Recall the language model objective (in this assignment we assume a bigram language model):

$$p(D) = \prod_{i=1}^{N} p(w_{i+1}|w_i) \tag{1}$$

For ease of presentation we will write this slightly differently:

$$p(D) = \prod_{(x,y)\in D} p(y|x) \tag{2}$$

where $D = (x, y)$ corresponds to all pairs of consecutive words in their order of appearance (y is the word which directly follows x) in the training corpus.

In the Word2Vec model we write:

$$p((x, y) \in D) = \frac{e^{x_o \cdot y_c}}{\sum_{z\in V} e^{x_0 \cdot z_c}} = p(y|x) \tag{3}$$

The goal of the model is the to find the optimal set of parameters θ^* such that:

$$\theta^* = \arg\max_{\theta} \sum_{(x,y)\in D} log(p(y|x)) \tag{4}$$

Let's start with some questions:

1. Write down the full term for θ^* as a function of the word vectors (i.e. integrate equation 3 into equation 4).
2. Using the above, write down the partial derivation of the objective according to its parameters by x_o.
3. Suppose that every word is represented with a 500 dimensional vector and that the vocabulary consists of 500K words.
   (a) How many parameters does the model have?
   (b) Is the model computationally feasible?

   Please support your answers with logical explanations.

As you have seen in the last question, the above model is not computationally tractable. [MSC+13] hence proposes a similar model where for every pair of words (x, y) we ask whether it appears in the training corpus (and hence $(x, y) \in D$) or not. Under this model for a pair of words that appear in the corpus they write, $D = 1$, and the probabilistic model is given utilizing the non-linear Sigmoid function:

$$p(D = 1|x, y; \theta) = \frac{1}{1 + e^{-x_o \cdot y_c}} = \sigma(x_o \cdot y_c) \tag{5}$$

For a pair (x, y) that does not appear in the corpus they write:

$$p(D = 0|x, y; \theta) = 1 - p(D = 1|x, y; \theta) \tag{6}$$

Trying to maximize the likelihood of the data is computationally intractable. This problem is solved by introducing a set of randomly sampled word pairs that do not appear in the training corpus: $D' = (x, y)$. This procedure is called **Negative Sampling**. The resulting objective is then:

$$\theta^* = \arg\max_{\theta} \sum_{(x,y)\in D} log(p(D = 1|x, y; \theta)) + \sum_{(x,y)\in D'} log(p(D = 0|x, y; \theta)) \tag{7}$$

This equation has a rather simple form and its gradients can be easily computed with the chain rule.

The final two questions are qualitative:

4. Can you give a non computational explanation as to why Negative Sampling should be helpful for a model?

5. The word representations induced by the model are used to enhance models that take into account word meaning. In lexical semantics we distinguish between the relation of **similarity** (for example the pairs *(car, vehicle) and (tiger, lion)* consist of similar words), and **association** (for example the pairs *(Maradona, football) and (star, sky)* consist of associated words). Would you expect words with similar Word2Vec vectors to be highly associated or highly similar?

References

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.