Assignment 9

Anusha Ramamurthy

PART A

Word2Vec is a tool that creates a vector representation of words in any given corpus. Representing words as vectors allows them to be used as features in many machine-learning applications. A word vector is a vector representation, we can think of it as a encoding, where we have N bits. N is the number of unique words in our corpus.

So the vector representation of the word is the vector in which the corresponding bit is one and all the others are zero. Assume for the sake of convenience, that we have four words "Dog, Cat, puppy, kitten" in our data set. Then Dog can be represented as "1000", Cat as "0100", puppy as "0010" and kitten as "0001" so on. This form of encoding is called One-hot encoding. However, it is not easy to identify word similarities with the representation we used above.

Instead, word2vec uses a distributed representation where each vector now is a weighted value, which is assigned, based on its relationship to other words in the corpus. Such a representation allows each vector to express the meaning of the word. This helps us to investigate questions of the form what king is to queen, like princess is to? It also helps in identifying a number of relationships like plurals, synonyms, countries and their capitals.

Word2Vec uses to learning models to learn from the word vectors, the bag-of-words model and the skip-gram model. A similar tool called GloVe is available. I found diverse opinions about the two tools. It appears that Word2Vec is a predictive model while GloVe is a count based one. The advantage of GloVe over Word2Vec is that GloVe can be parallelized and can train models much more easily.

PART B

Word2Vec is a neural network model to learn from text. It has wide scope. I found this method very unique and compelling. I wanted to get started by training it on a unlabeled data containing 50,000 sentences. The tool can be used to identify word similarities. Identify plurals, to identify the gender of a word, if it refers to its masculine counterpart or feminine. This was truly a puzzling concept to me. I was amazed that such powerful learning models exist. So I tested it on this set of sentences.

I dint not remove the stop words, as more words present in the training model, word2vec finds it easier to find associations. Had I more time; I would like to use it in sentiment analysis. It also seems to be a good way to build your own vocabulary. It could be used as a good and fast tool to identify synonyms, and logical reasoning. It can be used in an application to help students prepare for competitive exams like SAT, GRE etc.

We can see from examples that many times the similar words are inaccurate. We would need a labeled dataset in order to calculate the accuracy of the model.

PART C

I used the python wrapper gensim to implement word2vec. I found it quite difficult to set up and use. The main challenge is in identifying the input that the model takes. I found that it needs sentences in the form of list of list. The first line should contain the vocab size and the size of the vector.

I am relatively new to neural networks and it is quite complex to understand the inner workings of model. Especially changing the parameters for the model training, understanding how to provide the input seemed quite complex.

I found it quite confusing to tokenize the sentences. However I realized to my loss that Word2vec utilizes a lot of memory and has parallelism. Hence I had to install Cython and it utilized almost 100% of my laptops memory. I also had some errors with the encoding of the dataset I am using. Hence was able to parse around 30,000 sentences.

All in all it is a complex tool to use and needs a lot of reading before it can be used with a higher accuracy level for a task like sentiment analysis. A major disadvantage is the fact that the results only depend on the model used in training. If a new word is used in test the tool is unable to identify it and throws an error. Hence it needs a huge corpus, and training this will require a lot of GPU/CPU utilization. It does not seem like a tool to use on stand alone computer if one needs higher accuracy.

Implementation

Word2Vec.ipynb contains the implementation. Please be aware that it utilizes a lot of Memory and needs Cython instalation before run.