570 HW11: Word analogy task and Skip-gram Model

Daniel Campos dacampos@uw.edu

12/13/2018

1 Q3

3a: What is the "fake task" in order to learn word embeddings? That is, for this fake task, what are the input and the output at the **test** time?
The fake task is given a specific input word in the middle of a sentence(w1) and pick another word in the sentence at random what is the probability that the word is w2(for each word in the vocabulary). The input is a word in output is a probability distribution.

3b: How many layers are there in the neural network for solving the fake task? How many neurons are there in each layer?
Three layers in the network,input layer, a hidden layer and the output layer. Both input layer and output layer are the size of the vocabulary, the hidden layer is the dimensionality(50 in this example).

3c: Not counting the vector for the input word and the output vector for the output layer, how many matrices are there in the network? What are the dimensions of the matrices?
There are 2 weight matrixes one per layer first is 100,000 by 50 second is 50 by 100,000. How many model parameters are there? That is, how many weights need to be estimated during the training?
5,000,000 per layer so 10,000,000 weights

3d: Why do we need to create the fake task?
We use a fake task because we need a way to produce embeddings that approximate our data. If we didn't have the fake task we would have no way to get our embedding to approximate our goal.

3e: For any supervised learning algorithm, the training data is a set of (x, y) pairs: x is the input, y is the output. For the Skip-Gram model discussed in class, what is x? What is y?
The x is a word in the sentence and y is another word int he sentence with the window (+- words from x). Given a set of sentences, how to generate (x, y) pairs?
Starting at the beginning of sentence choose the first word w1 and then select the words withing the window before and after the word

each one of these will make a (x, y) pair. Do this for each position in the sentence.

3f: What is one-hot representation? Which layer is that used? Why is it called one-hot?
One-hot representation is a matrix representation of a word in vocabulary where all values are 0 except for the one representing the word, which is set to 1. This is used in the input layer and is called one-hot because it only has one hot bit(set to 1) and the rest are cold(set to 0).

3g: Softmax is used in the output layer. Why do we need to use softmax?
Softmax is used because it allows allows the output layer to be a probability distribution for the entire vocabulary. Instead of just providing an arbitrary number using softmax the output layer can provide a probability on how likely any word is to occur with the stopgram.