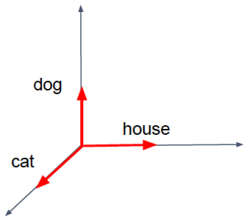
COMP 6721 Applied Artificial Intelligence (Fall 2023)

Worksheet #10: Deep Learning for NLP

Word Analogies. Word analogy questions often appear on standardized tests, like the SSAT, to test language aptitude and reasoning. Here's a simple one (fill in the blank): *Japan is to Sushi what Germany is to* ..... How can we solve this type of question with an AI? Stay tuned for the answer!

Word Vectors. Here are three words in one-hot vector representation (three words, so three dimensions):

'cat' = [0, 0, 1]
'dog' = [0, 1, 0]
'house' = [1, 0, 0]



What is the *distance* between the one-hot word vectors for (cat, dog) and (cat, house):

.....

Using the Euclidian distance,
$d(\vec{p}, \vec{q}) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$

Word2Vec: Training Data. Consider the following sentence: *"the cat drinks the milk"*. We will use this sentence to train a CBOW Word2Vec model. Assume that you use a context window of size 2 (1 word before and 1 word after the target word), and your vocabulary only contains the words in the sentence above.

Using only the sentence above, create the training instances using the CBOW method:

Instance	Context Word -1	Context Word +1	To Predict
1			
2			
3			

Word2Vec: Input Vectors. Now, (a) encode the vocabulary using one-hot vectors, assuming alphabetical ordering, no stop-word filtering (left) and (b) using these vectors, encode the three training instances above as input vectors for the network:

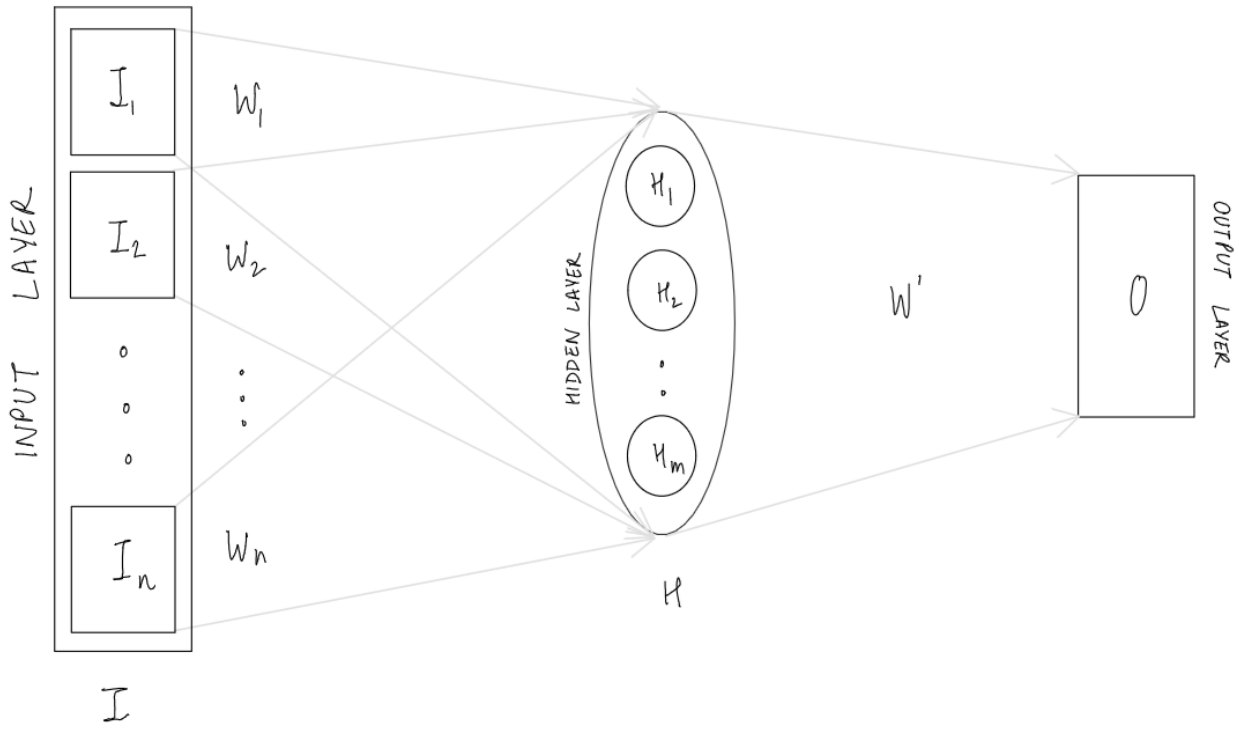
Word	One-Hot Vector			
cat				
drinks				
milk				
the				

Instance	Context	Word	One-Hot Vector			
1	Context Word -1					
	Context Word +1					
2	Context Word -1					
	Context Word +1					
3	Context Word -1					
	Context Word +1					

Word2Vec: Target Vectors. And what is the target one-hot vector for each training instance?

Instance	To Predict	One-Hot Vector			
1					
2					
3					

Word2Vec: Network. Assume that the Word2Vec model is trained with the network depicted below:



Assume that you want to produce word embeddings of dimension 2, using the training data you created above:

- What is the shape $(n \times m)$ of the input layer I? .....
- How many nodes m do we have in the hidden layer H? .....
- And the shape of the output layer O is? .....
- What's the shape of the weight matrix W from input to hidden layer? .....
- And the shape of the weight matrix W' from hidden to output layer is? .....

Note: (1) the Word2Vec network does not use a bias at either layer; (2) we apply one input word vector at a time and average the output of the hidden layer[1]

Softmax. Compute the *softmax* activation function σ on the vector v below:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \qquad v = \begin{bmatrix} 0.5 \\ 0.9 \\ 0.2 \end{bmatrix} \qquad \sigma(v) = \begin{bmatrix} ..... \\ ..... \\ ..... \end{bmatrix}$$

Solving Word Analogies. Ok, now re-write the question from the first task in form of a word vector calculation:

.....

[1] another approach is to average the input vectors before applying them