Here are some practice problems to work through in preparation for Exam 1.

Problem 1 (0 points) At some point in learning a byte pair encoding tokenizer as we saw in class, the following token pair:

$$(t_1, t_2) \tag{1}$$

has the maximal number of counts out of all pairs of tokens, and is added to the vocabulary as usual as the concatenation t_1t_2. When documents are now encoded with the new tokenizer for the next iteration of tokenizer training, can we see the pair (t_1, t_2) in the newly tokenized document? Why or why not?

Solution:

No. Consider a prefix in a document that so far has tokens not including t_1 or t_2. If the prefix continues as t_1 and then t_2, then it will be tokenized as t_1t2, so we come to a new prefix without t_1 or t_2. If the prefix continues as t_1 and then something other than t_2, it likewise won't result in a pair (t_1, t_2). Finally, if the prefix continues as something other than t_1, it also won't result in a pair (t_1, t_2).

Problem 2 (0 points) I've got the following neural network language model.

Consider a sequence $w_{1:t} \in \mathcal{V}^*$. We will overload each w as a one-hot vector in $\mathbb{R}^{|\mathcal{V}|}$. Let $E \in \mathbb{R}^{d \times |\mathcal{V}|}$ be an embedding matrix (learnable parameters.) Let $F \in \mathbb{R}^{d \times (dT)}$, that is, a matrix where the first axis is d dimensions, and the second is dT dimensions.

$$p(\cdot \mid w_{1:t}) = \text{softmax}(E^\top h) \tag{2}$$
$$h = \sigma(Fg) \tag{3}$$
$$g = [Ew_1; \cdots; Ew_{t-1}; \mathbf{0}_t \cdots; \mathbf{0}_T] \tag{4}$$

The definition of g here is a bit odd; we're using the $[\cdot;\cdot]$ notation to mean that we're concatenating the embeddings of all of our word embeddings up through $t-1$. Then we concatenate d-dimensional vectors of 0 values in the same way, up through T vectors, each of dimensionality d. This is to say that we concatenate the first $t-1$ word embeddings, and then for the future embeddings (those of the words we're trying to predict and those that could come after that,) we use the zero vectors so that our matrix shapes line up with F but we don't accidentally peak into the future.

Think about the expressivity limitations we've seen in lecture and in assignment 0. State similar expressivity constraints (types of word interactions, position information) of this model if there are any, and if not, explain why.

Solution:

There are no expressivity constraints like those we discussed in class. Positioning the words in the concatenation provides position information. The nonlinearity in the definition of h allows for nonlinear feature composition of all positioned words.

Problem 3 (0 points) I have matrices $G \in \mathbb{R}^{n \times m}$, $Q \in \mathbb{R}^{m \times p}$, $L \in \mathbb{R}^{d \times n}$. Write out the ordering of these matrices by which they can multiply.

Solution:

$$LGQ \tag{5}$$

Problem 4 (0 points) I have matrices $G \in \mathbb{R}^{n \times m}$, $Q \in \mathbb{R}^{m \times p}$, $L \in \mathbb{R}^{d \times n}$.

They can multiply in a single order (see above.) However, it's also the case that for matrices A, B, C that can multiply as ABC, one can compute their product as any of:

$$ABC = (AB)C = A(BC) \tag{6}$$

For the matrices G, Q, L, when they're multiplied in the order that's the solution to the problem above, what is the way of computing the product through two matrix multiplies (like $(AB)C$ vs $A(BC)$ by which one minimizes the number of scalar multiplications if we *further* assume that[1]

$$d = p \gg n > m \tag{7}$$

That is, d is equal to p, which are much greater than n, which is greater than m.

Solution:

$$(LG)Q \tag{8}$$

Because this results in dnm operations in the first multiply of L and G , and then dmd operations for the second multiply, of (LG) and Q. The alternative would be dmn for the multiply of G and Q, and then dnd for the multiply of L and (GQ).

Problem 5 (0 points) I have an embedding matrix $E \in \mathbb{R}^{d \times |\mathcal{V}|}$, and a neural network that provides representations h of prefixes $w_{1:t}$. One of my rows $E_{:,1}$ is equal to $2x$ of $E_{:,2}$. Another one of my rows $E_{:,3}$ is equal to $-E_{:,2}$. Probabilities in the network are modeled as:

$$P(\cdot \mid w_{<t}) = \text{softmax}(Eh) \tag{9}$$

What can we say about the relationships of the probabilities of words $1, 2, 3$ in this model? Specify which (all, or any subset) of words $1, 2, 3$ can be the maximum-probability word for *some* prefix $w_{1:t}$.

Solution:

The probability of word 1 will be low when the probability of word 3 is high, and vice-versa. The probability of word 2 will always be between that of 1 and 3. Words 1 and 3 can be the argmax, but word 2 can never. This is because its probability lies between that of 2 and 3. To see that these things are the case, look at the numerator of the softmax function for each of these terms. In any prefix, the denominator will be the same. One can compute the relationships given.

[1] While we're now setting $d = p$, don't use that to change the answer to the question above.