CS685 Quiz 2: *Transformers / transfer learning*
Released 3/1, due 3/7 on Gradescope (please upload a PDF!)
Please answer each questions in 2-4 sentences.

1. At training time, why can't we parallelize the computations of a recurrent neural language model? Why does the Transformer allow for this training parallelization?

Rnn's are sequential. So in order to compute a hidden state for the time t, we have to wait for the hidden state from t-1. To compute attention we need k, q, v which doesn't depend on the previous output. Hence we can process the entire input in parallel.

2. Give three reasons why we might prefer a character-based tokenization over a word-based tokenization when training a neural language model.

- We may need a smaller vocabulary size. In a language we may have infinite number of tokens. However number of characters are limited hence the vocabulary size is limited too. And smaller.

- If we have lots of spelling mistakes word tokenization will categorize them as unknown words. However if we use character based tokenization we can still get a meaning from them.

- It's simple to use. We don't have to deal with issues like "should we include punctuation" and so on. If you're working on a language that you don't know very well this can become handy.

3. Assume you are given a sequence-to-sequence model trained to solve the task of *paraphrase generation*, in which a model receives a single sentence as input and is asked to produce a paraphrase of that sentence (i.e., a sentence with approximately the same meaning). Why might you want to decode from this model using nucleus sampling instead of beam search?

Beam search chooses the most probable continuation path. If we're going to paraphrase it's very likely that similar words to the original sentence will be chosen. On the other hand nucleus sampling chooses from smallest possible set. This makes it much more likely that new words will be introduced to the generated sentence and it'll be more likely a paraphrase.