CS685 Quiz 3: *BERT / masked LM*
Released 9/28, due 10/1 on Gradescope (please upload a PDF!)
Please answer both questions in 2-4 sentences each.

1. **When pretraining BERT using the masked language modeling objective, we mask 15% of the tokens in each input sequence and then ask the model to predict the ground-truth identity of those tokens. What are the pros / cons of masking only 1% of input tokens instead? What about 99%?**

Answer:

The purpose of masking the input sequence during training is to hide parts of the input task so that the language model can attempt to learn what is below those masks while learning the surrounding context. If we mask 1% of the input tokens only, we would not be able to learn much about the text or check its performance, while if we mask 99% of the input, we wouldn't be able to learn much contextual information.

2. **Assume we have a training dataset for binary sentiment classification that contains only 10 labeled examples. We've also pretrained a Transformer language model on a huge amount of unlabeled data. Which of the following transfer learning setups would we expect to perform better in this scenario, and why?**
    a. **We design a deep recurrent neural network to classify the sentiment of a sentence. The inputs to this sentiment model are contextualized word embeddings of the labeled sentences, which are derived from the token-level representations at the final layer of our pretrained LM. Using our labeled sentiment dataset, we train the parameters of the sentiment model from scratch, while also fine-tuning the parameters of the LM.**
    b. **Instead of a deep RNN, our downstream sentiment model simply computes an element-wise average of the contextualized word embeddings from the pretrained LM. It then feeds these averaged vectors into a softmax layer for sentiment classification. Using our labeled sentiment dataset, we train the weight matrix of the softmax layer from scratch, and we do not fine-tune the LM's parameters.**

Answer:

B - As we only have a small number of training datapoints, it would not be possible to construct a deep RNN which can be expected to perform well. The second method should be more optimal in this scenario.