CS348K: Visual Computing Systems
Class/Reading Response Template

Reminder: please make sure the PDF you submit to Gradescope DOES NOT have your name on it. We will concatenate all responses and give everyone in the class a PDF of all responses.

Part 1: Top N takeaways from discussions in the last class. Note: this part of the response is unrelated to the current reading, but should pertain to the discussion of the prior reading in class (or just discussion in the class in general, if there was no reading):

- What was the most surprising/interesting thing you learned?
- Is there anything you feel passionate about (agreed with, disagreed with?) that you want to react to?
- Did class cause you to do any additional reading on your own? If so, what did you learn?
- Major takeaways in general?

Part 2: Answers/reactions to instructor's specific prompts for this reading. (Please see course website for prompts).

This recent paper applied a clever trick that allows the sequence of operations (a matrix multiply, a softmax, and then another matrix multiply) in an "attention" block to be able to be fused into a single memory-bandwidth efficient operation. The result of this trick is not only better performance, but that it became practical to run sequence models on larger sequences. As shown in the paper, the ability to provide longer sequences (wider context for reasoning) lead to higher quality model output.

Your job in your write is to address one thing. In your own words, describe how the algorithm works to correctly compute softmax(QK^T)V block by block. Specifically:

Start by making sure you understand the expression for computing softmax(x) for a vector x. Convince yourself that the basic formula suggestions you need to read all elements of the vector before you can compute the first element of softmax(x).

Now make sure you understand the factored form of the softmax, which is rwritten in terms of x1 and x2, where x1 and x2, are slices of the original vector x.

Now apply this understanding to the blocked algorithm. Line 9 of Algorithm 1 generates a partial result for a subblock of S_{ij}=(Q_i K_j^T). Then line 10 computes statistics of the rows of S_{ij}. Offer a high level explanation of why the subsequent lines correctly compute the desired result. The proof on page 23 in Appendix C works out all the math. Pretty cool, right?

The algorithm computes the softmax function for a large matrix product QK^T. The softmax function computes the exponential of each element of the matrix and then normalizes. The blocked algorithm divides the computation of QK^T into smaller blocks. For each block, it first computes a partial result S, which is the matrix product of the corresponding sub-blocks of Q and K^T. Then, it computes some statistics of the rows of S in line 10. Specifically, the algorithm first calculates the maximum value of each row in S and stores them in m~. Then, it subtracts m~ from each element of S, and compute the exponential result to obtain a new matrix P~. A high level explanation is the algorithm computes the softmax value in a sequential dataflow manner, where each step relies on the output of the previous step and successfully get the softmax of the whole matrix. If the maximum row value sunblock is greater than the value of the old subblock, then we update the new l using the computed l~ without maximum value and the old l devided by e^{maximum value}. This is actually a sum operation computed in a proportional way (both divided by e^{maximum value}), so that the result is still correct. This process is repeated for each subblock, and the final result is the sum of subblocks.

Part 3: [Optional] Questions I'd like to have specifically addressed via in class. (We also encourage you to just post these questions on Ed immediately so anyone can answer!)