TTIC 31230 Fundamentals of Deep Learning, winter 2019
Framework Problems

Problem 1: Consider the following softmax.

$$\begin{aligned} Z[b] &= \sum_j \exp(s[b,j]) \\ p[b,j] &= \exp(s[b,j])/Z[b] \end{aligned}$$

Give a back-propagation `+=` update based on the second equation for adding to s.grad using p.grad (and using the forward-computed tensors Z and s).

Give a back-propagation `+=` update based on the second equation for adding to Z.grad using p.grad (and using the forward-computed tensors s and Z).

Give a back-propagation `+=` update based on the first equation for adding to s.grad using $Z.grad$ (and using the forward-computed tensor s).

Problem 2: For the softmax in problem 1 show that we can instead use

$$\begin{aligned} e[b] &= \sum_j p[b,j]p.grad[b,j] \\ s.\text{grad}[b,j] &= p[b,j](p.\text{grad}[b,j] - e[b]) \end{aligned}$$

This formula shows how hand-written back-propagation methods for "layers" such as softmax can be more efficient than compiler-generated back-propagation code. While optimizing compilers can of course be written, one must keep in mind the trade-off between the abstraction level of the programming language and the efficiency of the generated code.