CS 6301.503 Spring 2019 Scott C. Waggener (scw180000)

Homework 3 (Calculus) Problem 1

Honor statement:

Read:

Solution: Complete ■

CS 6301.503 Spring 2019 Scott C. Waggener (scw180000)

Homework 3 (Calculus) Problem 2

Read:

Solution: Complete ■

CS 6301.503 Spring 2019 Scott C. Waggener (scw180000)

Homework 3 (Calculus) Problem 3

Read:

Solution: Complete ■

CS 6301.503 Spring 2019 Scott C. Waggener (scw180000)
Homework 3 (Calculus) Problem 4

Let $\boldsymbol{x}$ be the $K \times 1$ vector output of the last layer of a xNN and $e = \text{crossEntropy}(\boldsymbol{p}^*, \text{softMax}(\boldsymbol{x}))$ be the error where $\boldsymbol{p}^*$ is a $K \times 1$ vector with a 1 in position k^* representing the correct class and 0s elsewhere. Derive $\partial e / \partial \boldsymbol{x}$

Solution:

$$\frac{\partial e}{\partial \boldsymbol{x}} = ? \tag{1}$$

■

Proof: First, note that error function e is given by the cross entropy of a softmax, the typical error function chosen for **classification** networks. This establishes a $f(g(x))$ relationship wherein we must use the chain rule to differentiate e. Specifically, we will need to compute the following to apply the chain rule

$$\frac{d}{d\boldsymbol{x}}\text{softMax}(\boldsymbol{x}) \qquad\qquad \frac{\partial}{\partial \boldsymbol{p}}\text{crossEntropy}(\boldsymbol{p}^*, \boldsymbol{p}) \tag{2}$$

We will start with cross entropy. Recall that cross entropy is given by

$$\text{crossEntropy}(\boldsymbol{p}^*, \boldsymbol{p}) = -\sum_{x_i \in \boldsymbol{x}} p^*(x_i) \log p(x_i) \tag{3}$$

Differentiating cross entropy for each of $x_i \in \boldsymbol{x}$ produces a gradient vector. Recognizing that differentiating the summation in cross entropy with respect to a single x_i will produce a single nonzero term, we can say

$$\nabla_{\boldsymbol{x}}\Big(-\sum_{x_i \in \boldsymbol{x}} p^*(x_i) \log p(x_i)\Big) = \begin{pmatrix} -\frac{\partial}{\partial x_1} p^*(x_1) \log p(x_1) \\ -\frac{\partial}{\partial x_2} p^*(x_2) \log p(x_2) \\ \vdots \\ -\frac{\partial}{\partial x_k} p^*(x_k) \log p(x_k) \\ \vdots \\ -\frac{\partial}{\partial x_N} p^*(x_N) \log p(x_N) \end{pmatrix} \tag{4}$$

$$= \begin{pmatrix} 0 \\ 0 \\ \vdots \\ -\frac{1}{p(x_k)} \\ \vdots \\ 0 \end{pmatrix} \leftarrow k \tag{5}$$

So our cross entropy gradient for a one hot vector $\boldsymbol{p}^*$ with nonzero position k is 0 everywhere except at position k.

Now we compute our next required derivative: softmax The softmax function is given by

$$\text{softmax}(\boldsymbol{x}) = \frac{1}{\sum_{k=1}^{K} e^{x_k}} \begin{pmatrix} e^{x_0} \\ e^{x_1} \\ \vdots \\ e^{x_K} \end{pmatrix} \tag{6}$$

We need to derive the Jacobian of this function in denominator notation. □

CS 6301.503 Spring 2019 Scott C. Waggener (scw180000)
Homework 2 (Linear Algebra) Problem 2

(a) What is the arithmetic intensity for matrix matrix multiplication with sizes M, N, K.