CS 615 - Deep Learning

Assignment 1 - Forward Propagation
Winter 2024

Alec Peterson
ap3842@drexel.edu

1 Theory

1. Acknowledge teach of the following by **typing** your name after each of the following statements:

   (a) While you can use online resources, you may not copy them. (1pt) **Alec Peterson**

   (b) You cannot place any photos in your report (1pt) **Alec Peterson**

   (c) You cannot place any code snippets in your report (1pt). **Alec Peterson**

   (d) You cannot use any ML frameworks in your code, except for some purposes explicitly mentioned (1pt). **Alec Peterson**

   (e) While you may work in Jupyter notebook, or the like, you **must**, export your code as a python file for submission (and check that it works). (1pt) **Alec Peterson**

2. Given a single input observation $x = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$ and a fully connected layer with weights of $W = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$ as biases $b = \begin{bmatrix} -1 & 2 \end{bmatrix}$, what are the output of the fully connected layer given x as its input (5pts)?

   $h = [(1*1+2*3+3*5)-1, (1*2+2*4+3*6)+2]$
   $\implies h = [(1+6+15)-1, (2+8+18)+2)]$
   $\implies \mathbf{h = [21,\ 30]}$

3. Given an input, $h = [10, -1]$, what would be the output(s) if this data was processed by the following activation functions/layers (10pts)?

   (a) Linear: $g(h) = h \implies g(h) = [10, -1]$

   (b) ReLU: $g(h) = max(0, h) \implies g(h) = [10, 0]$

   (c) Sigmoid: $g(h) = \frac{1}{1+e^{-h}} \implies g(h) = [\frac{1}{1+e^{-10}}, \frac{1}{1+e^{1}}] = [0.9999546, 0.2689414]$

   (d) Hyperbolic Tangent:
   $g(h) = \frac{e^h - e^{-h}}{e^h + e^{-h}} = [\frac{e^{10}-e^{-10}}{e^{10}+e^{-10}}, \frac{e^{-1}-e^{1}}{e^{-1}+e^{1}}] = [1, 0.76159416]$

   (e) Softmax:
   $g(h) = \frac{e^h}{\sum_i e^{h_i}} = [\frac{e^{10}}{e^{10}+e^{-1}}, \frac{e^{-1}}{e^{10}+e^{-1}}] \approx [0.9999833, 0.0000167]$

2 Start Implementing Your Layers

See submitted code.

3 Connecting Layers and Forward Propagate

For following architecture:

Input→FC (1 output)→Logistic Sigmoid

Output result for first observation is: 0.49966372

See submitted code.