Author: Benjamin Smidt
Created: September 11, 2022
Last Updated: September 11, 2022

Assignment 2: Two Layer Neural Net

Note to the reader. This is my work for assignment two of Michigan's course EECS 498: Deep Learning for Computer Vision. This document is thoroughly researched but may not be perfect. If there's a typo or a correction needs to be made, feel free to email me at benjamin.smidt@utexas.edu so I can fix it. Thank you! I hope you find this document helpful.

1 Mathematics

2 Programming

2.1 NN Forward Pass

For this function we simply want to compute the forward pass through our two layer, fully connected network. Our input X has shape NxD, our first weight matrix W_1 shape DxH, first bias vector b_1 length H, second weight matrix W_2 shape HxC, and second bias vector b_2 length C. N is the number of examples, D the dimension of each example, H the number of layers in our hidden layer, and C the number of classes to classify our examples into.

Moving forward through the network is pretty simply. We begin by matrix multiplying XW_1 yielding the matrix H (for hidden) of shape NxH. Each column represents the computation that a given hidden layer does for *every single example*. Said differently, each row indicates the computations for *all the hidden layers* for a given example in N. Because of this we add our bias vector b_1 to every single row in our H matrix (using broadcasting).

Next, we compute our ReLU function using a mask and pass H off to W_2 to produce our final S matrix (for scores). Similar to our hidden layer computation, we matrix multiply HW_2 and add b_2 using broadcasting. This final S matrix has shape NxC as you would expect. Each example has a score for each class.

3 References

1. CS 231N: Neural Networks