Benjamin Smidt
6 September 2023

EECS498 A2: Linear Classifiers

This documents explains the mathematics and key ideas behind my solutions to *Assignment 2: Linear Classifiers* of Michigan's publicly available 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, please email me at benjamin.smidt@utexas.edu so I can fix it. Thank you! I hope you find this document helpful.

Contents

Linear Classifiers

Recall that our general approach requires to functions: a **score function** and a **loss function**. The score function takes the raw data (image pixels represented by a long vector) and outputs the predicted class scores. The loss function evaluates how well the scores predicted from our score function match the ground truth labels.

Definitions and Notation

We're using the CIFAR-10 dataset with $N = 50,000$ images in our training set. Each image has $3 \times 32 \times 32 = 3072$ pixels. Each image is 32×32 in size with 3 pixels at each location to indicate color and luminance (think RGB values).

We'll choose to represent each image as a 3072 dimensional vector where each index in the vector represents a single pixel value for that image. We'll denote this vector by$x_i \in \mathbb{R}^D$ where i indicates the particular image in the training set of images. Since the training set contains N images, $0 \leq i \leq N$. We'll denote the entire training set of images by the matrix $X \in \mathbb{R}^{N \times D}$, where row i of X is an image x_i. Since this is our training set, each image x_i has an associated label $y_i \in \{1, ..., K\}$ where K is the number of predefined categories the image could be classified into. For instance, $K = 1$ could indicate that the image is of a dog. $K = 2$ coud indicate the image is of a car, and so on.

Score Function

That is, the image has D number of pixels. In the CIFAR-10 dataset, an input x_i would have a dimension of

For a linear classifier it takes the form

$$f(x)$$

$$L_i = \sum_{j \neq y_i} max(0, s_j - s_{y_i} + \Delta)$$

$$\frac{\partial L_i}{\partial s_j} = \sum_{j \neq y_i} max(0, s_j - s_{y_i} + \Delta)$$

1 References

1. Softmax