Assignment 1 - Shape Classification

Bryon Kucharski

January 2019

1 A

For this section I used the numpy *histogram* function. Input variables are the number of bins, the range, and the option to convert to a probability distribution which is everything I needed for this question.

2 B

The learned parameters for logistic regression with L1-norm regularization is

w = [1.11530936e-04	-7.49933583e-05	-2.29022234e-04
-4.01947638e-04	-2.86620884e-04	-2.81880286e-04
-2.71110785e+00	-3.21798992e-01	1.55125088e+01
-4.92222327e+00]		

See Figure 1 for a plot of the loss

3 C

The output probabilities from the test set are

t = [0.92141292	0.16372102	0.18819288	0.89925192	0.20931707
0.05392417	0.16529335	0.87324538	0.9356319	0.05324894
0.9263984	0.79940756	0.04066315	0.9185448	0.88499432
0.26501827	0.90437085	0.9072313	0.18304947	0.0417109
0.21356196	0.52805257]			

The accuracy of my script is 100 percent

4 D

Without regularization, you can see that the weights are larger in magnitude.

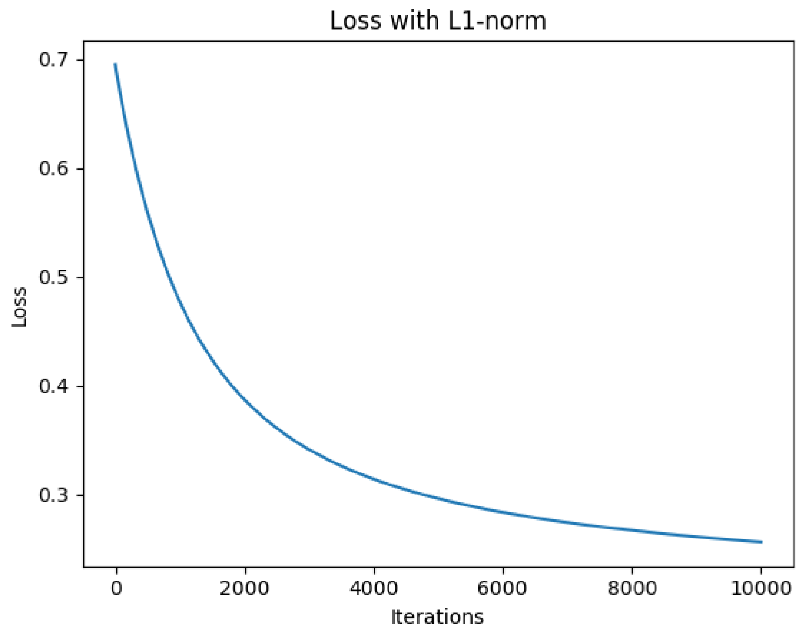

Figure 1: Loss with l1-norm

w = [0.17331586	0.14144335	0.35027896		
0.85988518	2.3034102	-3.15802307		
-6.1663501	-3.63369077	16.907654		
-7.99334777]				

t = [0.96190863	0.10108647	0.12155461	0.94770721	0.15122165
0.02399014	0.1057323	0.92921473	0.96861961	0.02644465
0.96254968	0.8568999	0.01941399	0.95966559	0.93825926
0.20644969	0.95144644	0.95403118	0.12486605	0.01971255
0.16244636	0.52460177]			

The accuracy of my script is still 100 percent. See figure 2 for a plot of the loss.

5 E

The equation I used for the cross-entropy loss with L1-norm regularization can be written as

$$y \log(\sigma(W * X)) + (1 - y) \log(1 - \sigma(W * X)) + \sum \mid W \mid \tag{1}$$

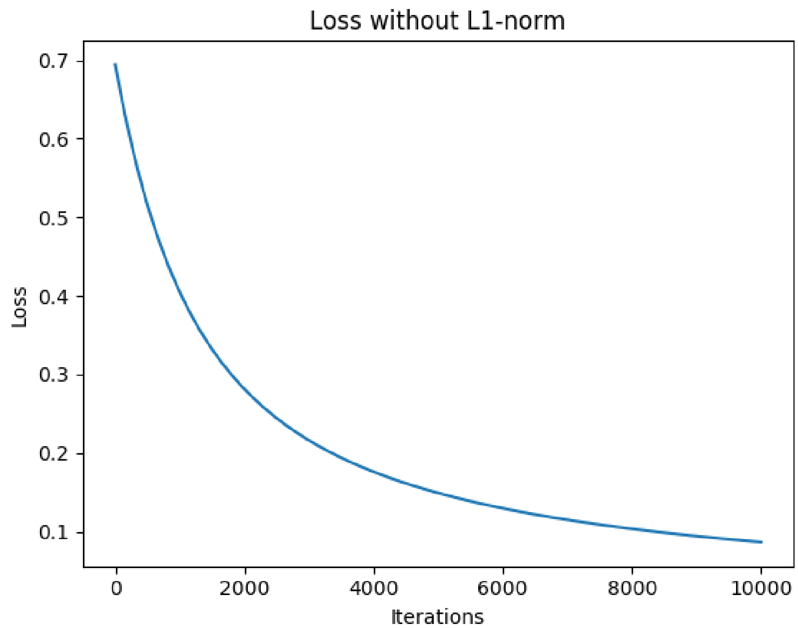

Figure 2: Loss with l1-norm

Where y is the true label, W is the weight matrix, and X is the input data. For now, I will ignore the L1-norm term.

$$y \log(\sigma(W * X)) + (1 - y) \log(1 - \sigma(W * X)) \tag{2}$$

Taking the derivative of the log terms yields

$$\left[\frac{1}{\sigma(W * X)} * y\right] * \frac{d}{dW}\sigma(X * W) + \left[\frac{1}{1 - \sigma(W * X)} * 1 - y\right] * \frac{d}{dW}\sigma(X * W) \tag{3}$$

Pull out the derivative term

$$\left[\frac{1}{\sigma(W * X)} * y + \frac{1}{1 - \sigma(W * X)} * 1 - y\right] * \frac{d}{dW}\sigma(X * W) \tag{4}$$

Taking the derivative of the sigma term yields

$$\frac{d}{dW}\sigma(X * W) = \sigma(W * X)(1 - \sigma(W * X))\frac{d}{dW}W * X \tag{5}$$

$$\frac{d}{dW}\sigma(X * W) = \sigma(W * X)(1 - \sigma(W * X))X \tag{6}$$

Plugging back into equation gives

$$\left[\frac{1}{\sigma(W * X)} * y + \frac{1}{1 - \sigma(W * X)} * 1 - y\right] * \sigma(W * X)(1 - \sigma(W * X))X \tag{7}$$

I will now focus on both the fractions inside the brackets. You can combine them into one fraction using algebra

$$\frac{y(1-\sigma(X*W)) + (1-y)\sigma(W*X)}{\sigma(W*X)(1-\sigma(W*X))} \tag{8}$$

distribute and simplify numerator

$$\frac{y - y\sigma(W*X) + \sigma(W*X) + y\sigma(W*X)}{\sigma(W*X)(1-\sigma(W*X))} \tag{9}$$

$$\frac{y + \sigma(W*X)}{\sigma(W*X)(1-\sigma(W*X))} \tag{10}$$

Plug back into original equation gives

$$\frac{y + \sigma(W*X)}{\sigma(W*X)(1-\sigma(W*X))} * \sigma(W*X)(1-\sigma(W*X))X \tag{11}$$

Cancel out terms to get

$$y + \sigma(W*X) * X \tag{12}$$

The only thing left is the l1-norm term. By definition

$$\mid x \mid = \frac{\mid x \mid}{x} \tag{13}$$

This is because if you have a negative number, it will have a negative slope. A positive number will still have a positive slope. The final equation can be written as

$$y + \sigma(W*X) * X + \frac{\mid x \mid}{x} \tag{14}$$

To achieve the desired values in my code, I used the numpy *sign* function, which turns all negative values to -1 and all positive values to 1. This is added to the derivative of the cross-entropy loss.