

Author: Benjamin Smidt
Created: September 7, 2022
Last Updated: September 9, 2022

Assignment 2: Softmax Classifier

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.

Loss

0.0.1 Mathematics

The equation for the loss function of a single example of Multinomial Logistic Regression is:

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_{j=1}^C e^{f_j}}\right) = -f_{y_i} + \log\left(\sum_j e^{f_j}\right) \quad (1)$$

$$f_j = f(x_i, W)_j = (Wx_i)_j \quad (2)$$

Thus, to find the loss for the training data, we simply need to average the loss L_i for each example. For SVM's, the loss was called *hinge loss*. The loss for a Softmax Classifier is known as *cross-entropy loss*.

0.0.2 Programming

In our implementation of finding the loss, we first matrix multiply XW and raise every value in the resulting matrix to e giving us matrix e^f which has shape $N \times C$ since X and W have shapes $N \times D$ and $D \times C$. (N = examples, D = feature dimensions, C = classes). ADD VECTORIZES IMPLEMENTATION HERE.

To check our implementation, we do a quick sanity check. Since we initialized all our weights w_i to be very close to 0, we expect $e^{f_{y_i}}$ to be

one-tenth of $\sum_j e^{f_j}$ since $e^0 = 1$. Since $-\log(1/10) = \log(10)$, $\log(10)$ is approximately our expected loss value.

0.1 Gradient

0.1.1 Mathematics

To find the gradient with respect to our weight matrix W , let's again focus on one example. We can rewrite our loss function as:

$$L_i = -\log\left(\frac{e^{W_{y_i}x_i}}{\sum_j e^{W_jx_i}}\right) \quad (3)$$

where W_{y_i} represents the weights for the correct label for example i and W_j is the weights for any given class (including W_{y_i}) for example i . Note that since the shape of W is $D \times C$, each W_j is a column of W .

Let's start by reformulating our loss function a bit.

$$L_i = -\log\left(\frac{e^{W_{y_i}x_i}}{\sum_{j=1}^C e^{W_jx_i}}\right) \quad (4)$$

$$L_i = \log\left(\frac{\sum_{j=1}^C e^{W_jx_i}}{e^{W_{y_i}x_i}}\right) \quad (5)$$

$$L_i = \log\left(\sum_{j=1}^C e^{W_jx_i}\right) - W_{y_i}x_i \quad (6)$$

Then we find the gradient with respect to W_{y_i}

$$\frac{\partial L_i}{\partial W_{y_i}} = \frac{\partial}{\partial W_{y_i}} \log\left(\sum_{j=1}^C e^{W_jx_i}\right) - \frac{\partial}{\partial W_{y_i}} W_{y_i}x_i \quad (7)$$

$$\frac{\partial L_i}{\partial W_{y_i}} = \frac{\partial}{\partial W_{y_i}} \log\left(\sum_{j=1}^C e^{W_jx_i}\right) - x_i \quad (8)$$

$$\frac{\partial L_i}{\partial W_{y_i}} = \frac{1}{\sum_{j=1}^C e^{W_jx_i}} \frac{\partial}{\partial W_{y_i}} (e^{W_1x_i} + \dots + e^{W_{y_i}x_i} + \dots + e^{W_Cx_i}) - x_i \quad (9)$$

$$\frac{\partial L_i}{\partial W_{y_i}} = \frac{x_i e^{W_{y_i}x_i}}{\sum_{j=1}^C e^{W_jx_i}} - x_i \quad (10)$$

$$\frac{\partial L_i}{\partial W_{y_i}} = x_i \left(\frac{e^{W_{y_i} x_i}}{\sum_{j=1}^C e^{W_j x_i}} - 1 \right) \quad (11)$$

The math works out similarly for the gradient with respect to W_j

$$\frac{\partial L_i}{\partial W_j} = \frac{\partial}{\partial W_j} \log \left(\sum_{j=1}^C e^{W_j x_i} \right) - W_{y_i} x_i \quad (12)$$

$$\frac{\partial L_i}{\partial W_j} = \frac{\partial}{\partial W_j} \log \left(\sum_{j=1}^C e^{W_j x_i} \right) \quad (13)$$

$$\frac{\partial L_i}{\partial W_j} = \frac{1}{\sum_{j=1}^C e^{W_j x_i}} \frac{\partial}{\partial W_j} (e^{W_1 x_i} + \dots + e^{W_j x_i} + \dots + e^{W_C x_i}) \quad (14)$$

$$\frac{\partial L_i}{\partial W_j} = \frac{x_i e^{W_j x_i}}{\sum_{j=1}^C e^{W_j x_i}} \quad (15)$$

0.1.2 Code

As before, we implement the loss and the gradient in the same function to save computation.

1 References