Assignment 2

Softmax Classifier

Loss

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

$$L_i = -log(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}) = -f_{y_i} + log(\sum_j e^{f_j}) \tag{1}$$

$$f_{y_i} = f(x_i; W) = Wx_i \tag{2}$$

Thus, to find the loss for the training data, we simply need to average the loss L_i for each example.

0.0.1 Naive Implementation

In our Naive 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 NxC since X and W have shapes NxD and DxC. (N = examples, D = feature dimensions, C = classes). We then use nested for loops to sum e^{f_j} in each class and divide the $e^{f_{y_i}}$ by that sum. Note that $e^{f_{y_i}}$ is included in e^{f_j}. Finally, we take the $-log()$ of that result giving us our individual loss. We iteratively sum all those losses and divide by N to find the average loss.

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.

0.0.2 Gradient

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:

$$L_i = -log(\frac{e^{W_{y_i}x_i}}{\sum_j e^{W_j x_i}}) \tag{3}$$

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 DxC, each W_j is a column of W.

From here we can compute the gradient with