Amber Nelson
nels9242
Homework 4

get_mini_batch
First this function uses random.sample over all possible images indices given a batch_size. All of the indices returned from this function are unique. Then I have for loops that build all of the batches so that it returns mini_batch_x of size (batch_size x batch_size x 196) and mini_batch_y of corresponding labels of size (batch_size x batch_size x 10). I had to go back to this function a few times to make sure it was working correctly. I encountered a pretty bad problem where the way it was storing the images was conflicting with how all of my other code was retrieving them (it was training the weights and bias for scrambled image data). This was fixed pretty easily once I found the weird issue.

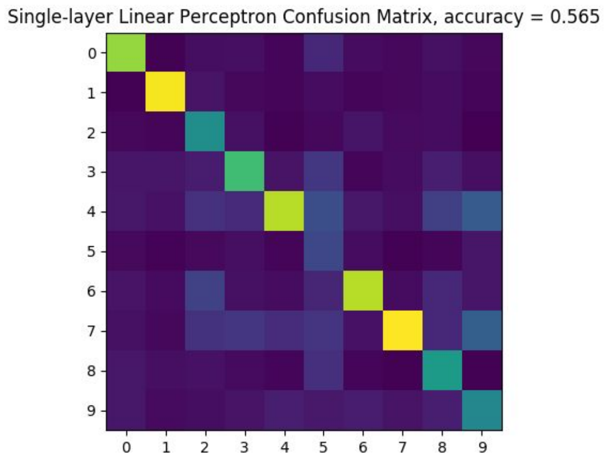
fc
I had some problems with how x in this function was being represented in order to make the dot product work. That is why I ended up including some basic reshaping code for x (for example, from shape (10, 1) to (10,)). This function just computes y = wx + b.

fc_backward
For some reason I really struggled to wrap my head around the derivatives here. I think I just needed some calc review since its been a while since I've really had to do a lot of derivatives. resI also ended up including some code to reshape dl_dy and x if their shapes caused issues.
Given dl_dy:

dl_dx = dl_dy * w
dl_dw = dl_dy * transpose(x)
dl_db = dl_dy

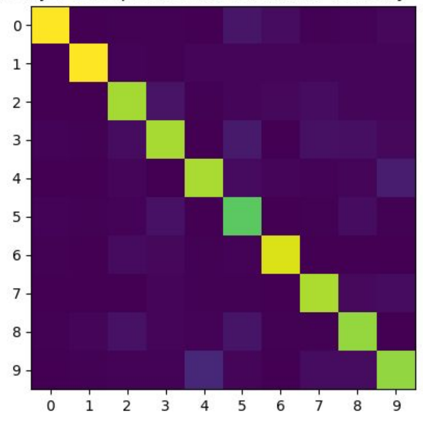
loss_euclidean
This function was very simple to implement. Here, dl_dy = -2*(y - y_tilde).

train_slp_linear
This function was difficult to get working, mostly because I couldn't figure out how the parameters should be tuned at first. It performs very well now given 100 iterations and a learning rate of 0.005 (it usually achieves over 50% accuracy with this).



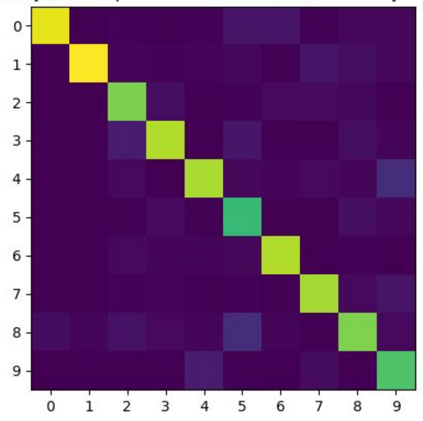
loss_cross_entropy_softmax
I had a lot of trouble with this function in just getting it to stop causing number to underflow/overflow. This problem didn't really ever happen in implementing train_slp but became very evident when implementing train_mlp. I ended up re-ordering how I was computing the loss and y_tilde here to try and retain better numerical precision. Also, calculating the derivative for this was a lot more complicated than the derivatives in the previous functions. I found out that the result was actually super simple though. dl_dy = y_tilde - y.

train_slp
The loss derivative was pretty difficult to derive for this one, but this function overall was not too much more complicated than train_slp_linear. I was able to get it to consistently perform with at least 82% accuracy with 100 iterations and a learning rate of 0.05.

Single-layer Perceptron Confusion Matrix, accuracy = 0.859


relu
The only issues I ended up finding with relu was in the shape of the input x. I did not really make it that versatile to handle variation in the shape of the input matrix x. Otherwise, relu was simple to implement.

relu_backward
I'm a little uncertain as to if this was implemented correctly. I have it compute dl_dx using the convention that if x_i is greater than 0, its added to the ith spot in dl_dx. Else, its set it 0. Then to get the final dl_dx, I multiply dl_dy with dl_dx.

Multi-layer Perceptron Confusion Matrix, accuracy = 0.836


train_mlp
I had a lot of trouble figuring out if I was actually implementing this right. I basically implementing it like the following:

y = fc(relu(fc(image, w1, b1)), w2, b2)

The output y was passed into loss_cross_entropy_softmax, then back-propagation was one from there. I struggled to get anything above 85% accuracy with this, so something was definitely incorrect somewhere, but I was not able to debug it enough to figure out where the problem was.

conv
Despite not implementing train_cnn, I was able to implement this function. It was hard really understanding at first the order things were all happening in but I did successfully get this function working.

pool2x2
I also implemented this function, which was not all that difficult.

I did not have time to complete the rest of the homework from here. Despite starting these homeworks 2 weeks out from the deadline, I *really struggle* to get them completed in time. I always end up spending as much waking time I can in the last 5 days just working on these homeworks. If I had more time I'm sure I could complete them but I just haven't been able to do it with the time we've had for each.