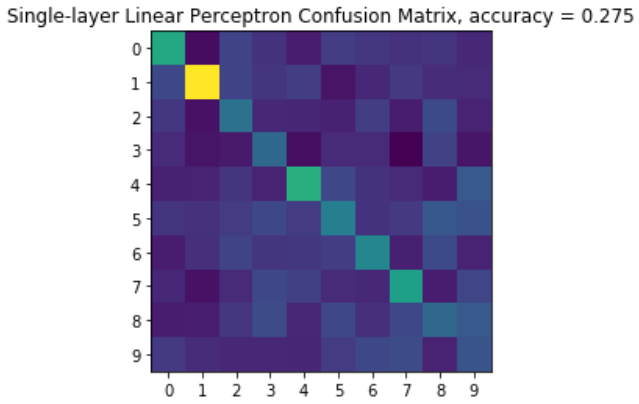
David Ma

Computer Vision

HW 4: Convolutional Neural Network

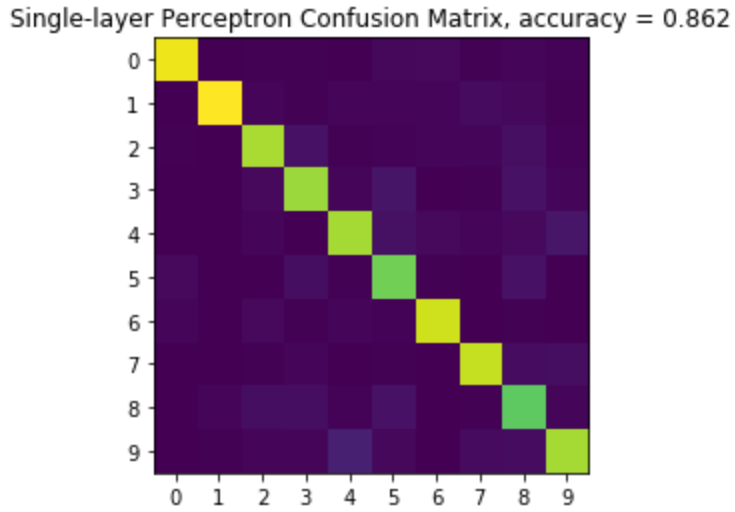
1. SLP Linear

This part was pretty simple, and I was able to easily get an accuracy of more than .25, as shown below. The functions were pretty straightforward, and there's nothing much that I need to say about them.



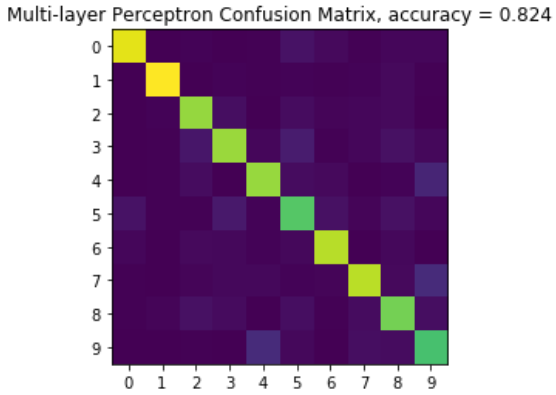
2. SLP

This one was a little more difficult, but I was able to get more than .85 after tuning hyperparameters. The auxiliary functions themselves weren't difficult after the math was all said and done.



3. Multi-Layer Perceptron.

This is where things started getting difficult. relu and relu_backward are both really simple functions, so I don't think the problem was in either of those two. At the same time, though, I emailed Jae Shin Yoon and he said my structure of train_mlp looked correct, so I think it was a hyperparameter problem. At the end of the day, I just wasn't able to find the hyperparameters to make it work, and my accuracy ended up just higher than .8.



4. Convolutional Neural Network

This was the most complex part, and my code really stopped working on this one. I implemented conv and conv_backwards with im2col, but I also included alternate implementations that rely heavily on for loops. flattening and pooling were both pretty simple so I don't think the problem was there; so I think I either have a bug in my train_cnn or in my conv implementations. Either way, I wasn't able to get even close to the correct accuracy. I think my overall structure is correct, though, so I'm hoping for partial credit. Additionally, you can see that the loss function is decreasing, as shown on the right.