Devin Liu
<hideUID>

HW #9

1. I modified line 70 and 71 to answer this question. I changed mnist.test.images and mnist.test.labels to mnist.train.images and mnist.train.images respectively. The accuracy I got in the train set was somewhere between 91% and 92%, but averaged around 91.5% after multiple runs. There wasn't really a huge difference in the accuracy between the test and training set as the test set had around a 92% success rate. However if I was to explain this 0.5% of a difference, I would point to the difference in sample size. The test set has 10000 points of data while the training set has 55000 points of data. It could very well be the case that the test set had too small of a sample size for the success rate to regress to the mean of 91.5%.
2. After changing the number from 1000 to 10, I got an accuracy of exactly 78.21%. After changing the number from 1000 to 10000, I got an accuracy of about 92.4%. The dip in the accuracy in the first case occurred because I reduced the amount of training that the model will go through. As a result, its accuracy suffered because it had less data points to learn from. In the second case, the accuracy increased because there were more data points to work with.
3. After the change, the accuracy remained around 91.8% to 91.9%, which is not significantly different than the 92% accuracy before the changes. The reason there isn't much of a difference is because these W and b values that get initialized most likely won't be the final values. This is because the algorithm will work towards new values for W and b that will work better in recognizing the correct digits. As a result, the difference in initialized values for W and b won't make a huge difference since W and b will be altered anyways.