For lab 3 you experimented with several methods to improve the accuracy of classification using a small training set for the MNIST task. For this lab, your task is the same as before, but now use convolutional neural networks implemented in tensorflow.

As before, use the first 500 examples in the xtrain.txt file as your training set, leaving the test set unchanged. Try to obtain the highest accuracy possible using the following techniques, which have been explained in class. If a particular technique is provided in tensorflow, you don’t need to re-implement it.

1. Training set augmentation
2. Test set augmentation and testing by committee
3. $L_2$ regularization
4. Momentum
5. Optimizing parameters (learning rate, hidden layer sizes)
6. Dropout
7. Adversarial training
8. Adaptive learning rate
9. Using the remainder of the training set as unlabeled data

Your grade will be based on the number of techniques you implement correctly and on the quality of the results you obtain. As usual, write a report describing your work. Include plots of the loss function and accuracy as training progresses (per epoch, not per batch, for both training and test sets). Comment on the benefits provided by each technique, particularly in relation to the increase (if any) in running time that they entail and the difficulty in implementing them. For extra credit, experiment with more extreme versions of this problem. What if we only have 50 training examples? What if we only have one training example from every class?