Submission instructions:

• If a problem asks for a numerical answer, you need only provide this answer. There is no need to show your work, unless you would like to.

• Please type up your solutions. We suggest using an online latex editor like www.overleaf.com, though this is not a requirement.

• Upload the PDF file for your homework to gradescope by midnight on Friday, Mar 22.

This homework will be focused on designing your own models for digit classification. You will be conducting experiments under your choice of two conditions: (1) where your model has access to the full set of 60K labeled MNIST images, and (2) where you model only has access to 6k labeled MNIST images, but can make use of as many unlabeled images as is helpful. The training sets can be directly downloaded from the corresponding Kaggle competitions – instructions for gaining access are posted on Piazza. You will turn in your final write-up, described below, as well as make at least one Kaggle competition submission, also described below. For this homework, you only need to choose one of the following two options to complete and write up. However, for extra credit, you may complete both or do something more substantial for either option.

(1) For the first option, you will be building a supervised digit classifier on a moderately large training set consisting of 60k labeled images. This problem is open-ended, and you may apply any techniques or knowledge we have discussed throughout the course. While we will assign extra credit based on the depth of your investigation, novelty of your models, and accuracy of your best system, the only requirements are:

• You must run experiments with at least two different systems and draw comparisons between their performance and the types of errors they make. We expect you two write at least two paragraphs describing the systems you implemented, including the hyperparameters you found that worked best, and including at least two plots (e.g. validation vs training accuracy per epoch).

• You must create at least one submission to the supervised Kaggle competition using one of your systems.

Examples of plausible projects for this part include, for example: comparing fully connected neural nets with convolutional neural nets, comparing deep convolutional neural nets with shallow ones, comparing the effects of different types of regularization on neural models, etc..

Examples of going above and beyond include, for example: creating complex and finely-tuned architectures that achieve substantial performance gains, doing an in-depth evaluation of the effects of max-pooling layers, creating your own type of layer for a neural model, implementing new and recent advances in neural nets by reading papers not discussed in class (e.g. attention layers, or recurrent layers), conducting an in-depth analysis of the types of errors different models suffer from, comparing more advanced regularization techniques like dropout with conventional ones like L2 and L1, etc..
(2) For the second option, you will be building a semi-supervised digit classifier on a smaller labeled training set consisting of 6k labeled images, in addition to all the input images (without labels) from the larger 60k training set. This problem is open-ended, and you may apply any techniques or knowledge we have discussed throughout the course. While we will assign extra credit based on the depth of your investigation, novelty of your models, and accuracy of your best system, the only requirements are:

- You must at least implement PCA for learning a lower-dimensional feature space from the unlabeled data.
- You must at least train a supervised neural classifier (either fully connected or convolutional) on the 6k labeled set that uses your lower-dimensional feature space as input.
- We expect you to write at least two paragraphs describing your approach to semi-supervised learning as well as your results, including the hyperparameters you found that worked best (e.g. dimension of PCA projection), and at least two plots (e.g. validation accuracy as a function of dimensionality of your learned feature space).
- You must create at least one submission to the semi-supervised Kaggle competition using your system.

Examples of plausible projects for this part include, for example: exploring the effect of dimensionality reduction on training and validation accuracy, exploring the effect of size of unlabeled data vs. size of labeled data, etc.

Examples of going above and beyond include, for example: implementing more advanced dimensionality reduction techniques like autoencoders, implementing new and recent advances in semi-supervised learning by reading papers not discussed in class (e.g. variational autoencoders, self-training), doing an in-depth analysis of how dimensionality reduction and model architecture interact to determine generalization performance, etc...