-----NEURAL NETWORK CODE-----

-----ANALYSIS CODE-----

**get\_ensemble\_confusionMat.m**

Gets the mean confusion matrix and the standard deviation of each entry for an ensemble of networks given a directory that contains them.

**get\_nn\_confusionMat.m**

Gets the confusion matrix for a single network.

**get\_nn\_filters.m**

Creates an image of 64 filters from the first hidden layer.

**get\_nn\_mapping\_vector.m**

Creates a vector by concatenating the softmax outputs of a network for each of 10 examples in every digit class to be used as a representation of its input-output mapping.

**get\_nns\_mapping\_matrix.m**

Compiles a matrix of mapping vectors by looping through networks in a given directory and calling get\_nn\_mapping\_vector().

**graph\_nn\_corruption.m**

Graphs the 2D tSNE visualization of networks of different corruption types given a list of directories that are used to call get\_nns\_mapping\_matrix().

**plot\_nn\_corrupts\_err.m**

Plots test error over time for networks of different corruption types.

**plot\_nn\_err.m**

Plots training and test error over time for a single network.

**visualize\_confusionMat.m**

Emphasizes parts of confusion matrix that were wrong by taking the log of a constant times its values and resizing the image to five times its original scale.

**visualize\_nn\_weights.m**

Attempts to visualize weights of upper layers (as well as the first layer by calling get\_nn\_filters()) by taking the average of images in the training set weighted by the activations of each hidden unit when that example is being fed forward through the network. However, the resulting filters are not sufficiently different to make a reliable comparison using this method.