

Week 1 Description

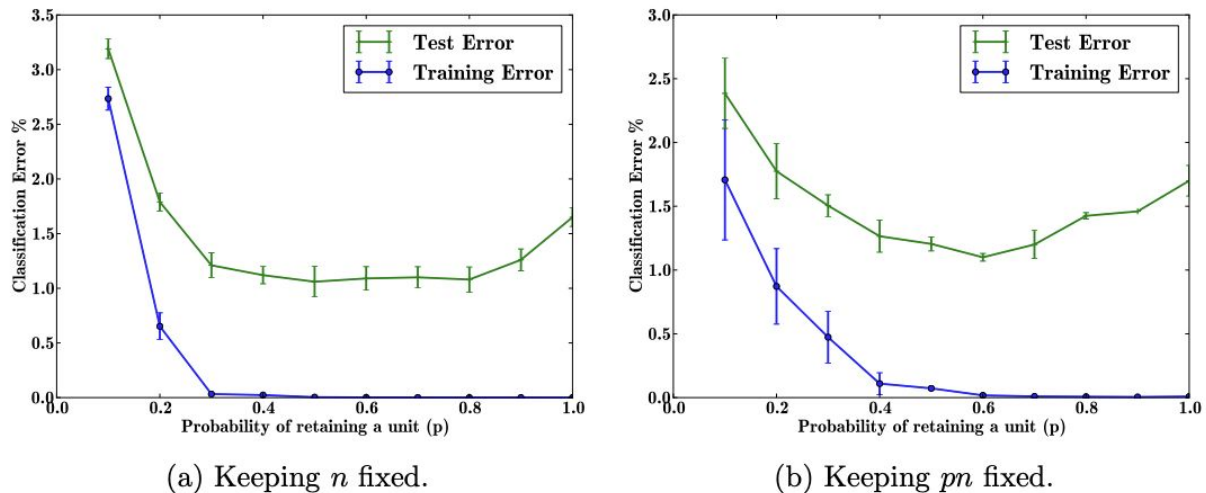


Figure 9: Effect of changing dropout rates on MNIST.

Figure Referenced from: Srivastava, N., Hinton, G., Krizhevsky, A., Krizhevsky, I., Salakhutdinov, R., Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Figure 9

Problem Setting:

With limited training data, many complicated relationships between inputs/outputs will be a result of sampling noise. They will exist in the training set, but not in real test data even if it is drawn from the same distribution. This complication leads to overfitting, this is one of the algorithms to help prevent it from occurring. The input for this figure is a dataset of handwritten digits, and the output after adding dropout are different values that describe the outcome of applying the dropout method. All in all, less error is outcome after adding dropout.

Data Sources:

A real world problem that this can apply to is google searching, someone may be searching for a movie title but they might only be looking for images because they are more visual learners. So dropping out the textual parts, or brief explanations will help you focus on the image features. The article states where they retrieve the data from (<http://yann.lecun.com/exdb/mnist/>). Each image is a 28x28 digit representation. The y labels seem to be the image data columns.

Algorithm:

My goal in reproducing this figure is to test/train the data and calculate the classification error for each probability of p (probability of retaining a unit in the network). My goal is to get p to

increase as the error goes down to show that my implementation is valid, and I will tune this hyperparameter to get the same outcome. I will do this by looping through all the training and testing data using a 784-2048-2048-2048-10 architecture and keep the n fixed then change p_n to be fixed. I will then gather/write the data into a csv file. This csv file will then contain all the necessary data to output the figures. In this project, I will learn how the dropout rate can benefit the overall error in a neural network.