Computer Lab, April 23

Problem 3 modified slightly Apr 24. Lots of Python and Matlab examples were uploaded.

In this homework, we'll look at using linear neural networks and RBFs for classification and regression.

Problem 1: Linear Network for Classification

Using the classification code from Apr 14, 16 as templates, create a linear classifier for the wine data (using all at once training with the pseudoinverse- Please build your pseudoinverse manually with the SVD). Show your results using the confusion matrix. Use online training (Widrow-Hoff), with the "TGF" example as a starting point.

Problem 2: RBF Experiment- Orthogonal Least Squares

In this problem, we want to use "orthogonal least squares" to locate cluster centers. The code used for the wine data (on the class video) is also provided.

The data file

We'll be using the Pima Indian Diabetes database from the earlier work (write up of the data is in the folder from Apr 14). To summarize, the data will be in the matrix X, which willbe 768×8, T is 768×2 (these are the targets defined as column 1 or column 2 of the 2×2 identity), and `targets` is the class label as "1" or "2".

The RBF

Use the given code from the wine experiment to complete the diabetes classification data. Be sure to give all of the relevant information about your RBF (especially number of centers).

Problem 3: RBF Experiment- Clustering

In this problem, we want to use data clustering to find the location of the centers. We can also set a desired radial spread for the Gaussians (if you choose to use Gaussians). The data will be the same set we used before in the "digits lab". You'll recall that we've already performed a data clustering, and the results looked pretty good (using 10 centers).

The data file

When you load `handdigits.mat`, you'll see matrix X that is $64x1797$, targets T is 10×1797, and `targets` is a vector that has the scalar values of the classes, 1 to 10 (rather than 0 to 9).

Python users might want to load the Matlab data file, otherwise, you may have to build the matrix T (you might double check that). There's a sample given in the 2d problem).

The RBF

We'll use those 10 centers for our RBF classifier. Start by randomly splitting the data into 70% for training and 30% for testing. Training will consist of computing the $k-$means centers, and then solving for the weights and biases. The sample Matlab and Python files will show you how to use k-means for regression, so you'll need to modify those for classification (and creating/printing the confusion matrix).

What to turn in

Please write up your results as we did in the k-nearest neighbor lab. Include relevant graphs and code snippets. You can collate all three problems into one document if you wish- We'll be flexible with the section headings. Mainly, this will be a nice way of summarizing and presenting your work in one package.

Due: One week from Apr 23 (so that would be Apr 30).