

Modeling Lab (Due Wed, Oct 24)

This week, we're building linear neural networks using either online training (training where the data is not all known at once) using the Widrow-Hoff rule, or batch training, where the data is all known at once. In the latter case, we can solve the linear equation using the SVD (form the pseudoinverse).

Classification Problem 1: The Iris Data

In this example, each vector has 4 sample measurements from one of 3 classes of flower. There are 50 flowers in each class, so as a matrix, the inputs X are 150×4 (load the data from the file online, `IrisDataX.mat`).

The “targets” tell us which class the flower is, and we have to build this set. As we did in class, each class is a column of the 3×3 identity matrix (we have 3 classes). Thus, write the code that creates the output matrix T (it is 150×3).

Once you have the target data, build the linear neural network using the Widrow-Hoff online training rule (download that from our class website). You might experiment a bit with the learning rate, but you might start with something small like $\alpha = 0.01$.

Finish the problem by plotting your error (`EpochErr` is the output), and then print the confusion matrix.

Classification Problem 2: Breast Cancer Data

The data is given as an m -file named `BreastData.m` (so in the command window, to load the data, type `BreastData` (without the $.m$ suffix).

Each data “point” represents nine measurements taken from a breast exam, and there are six target classes:

- | | | |
|------------------|---------------|---------------|
| 1. Carcinoma | 3. Mastopathy | 5. Connective |
| 2. Fibro-adenoma | 4. Glandular | 6. Adipose |

Therefore, our mapping is go from \mathbb{R}^9 to \mathbb{R}^6 .

If you download the Matlab m -file (this is NOT a binary data file, but a plain text script), you can see a few notes there. To load the data into Matlab, just type `BreastData` in the command window (or in a script) to load the matrices X and T . Notice that they may not be given to you with the right dimensions (the data may be in columns or in rows), so be sure to check that and use the dimensions appropriate for your algorithm.

Using this data, construct **two script files**- One using the online-training and one that uses batch training (all the data at once) and the pseudoinverse (we should compute the pseudoinverse “by hand” using the SVD!)

Both scripts should “output” the final confusion matrix.