Linear Network Classifier Example and HW

- First take a look at the "TGF" classifier that we went through in class. In particular, notice that the data being input is in X and is 16×6 (meaning 6 data points in 16 dimensions). The target data is in T and is 3×6, meaning 6 points in 3 dimensions (3 because there are 3 classes).
- In this example, Widrow-Hoff is used to train the network. Please pay attention to the dimensions of the inputs, the outputs, the weights W and the vector of biases b.
- Line 35 shows you how to construct the output Y given data in X, and the weights and biases.
- The last three lines show you how to convert the vector outputs for classification into one dimensional values to put into the confusion matrix (You shouldn't need to change those).
- **We only had 6 points, so we didn't separate the data. You should separate the data into training and testing sets below.**
- Training the data "all at once" happens in the file `TGFExample2.m`, where the pseudoinverse is used.

Linear Network Classifier Homework

We're going to work with some data gathered from breast exams. The data represents 106 patients with 9 measurements each, and the output is a classification (integers 1 to 6). The data is stored as a text file in `BreastData.m`, so you can look at the file for more information about where the data comes from.

To load the data, just type `BreastData` and a matrix X (that is 106×9 and a target matrix T (that is 6×106) is loaded. Try to be sure that the dimensions are what you need them to be for each of the algorithms.

For the lab,

1. Split the data into Training and Testing sets (70% for training, 30% for testing).
2. Do some data exploration. If you think you need to scale the data, you can use `StandardScaler.m`.
3. We'll build two models- one using Widrow Hoff, and one using "batch" with the pseudoinverse.
4. At the end, please note the confusion matrix output and the overall error.