

Modeling, HW 15

This homework will not be turned in, but you should work through it before working on the final exam.

Goal:

Create a neural network that can identify written characters (it can read!).

Details

1. Download the data from the class website. It contains a matrix **alphabet** that is 35×26 , one column for each letter. Each column contains 35 numbers- This is an array that is 7×5 , and the numbers can be thought of as grayscale values. For example, in class we saw that:

```
load alphachars
h=alphabet(:,1);
imagesc(reshape(h,5,7)');
axis equal; axis off;
```

The array gives you 26 inputs, one for each character- That's not enough for good training. You can build more samples of each character yourself. For example, in class, we said that:

```
X=[alphabet alphabet+0.1*randn(size(alphabet))];
```

doubles your training set. You should have at least 16 samples of each character- with more and more noise. The constant before **randn** will change the amount of variance in the noise you add, and go ahead and make it range from 0.01 to 0.6. For example,

```
X=alphabet;
T=eye(26);
noise=linspace(0.01,0.6,15);
for j=1:15
    X=[X alphabet+noise(j)*randn(size(alphabet))];
    T=[T eye(26)];
end
```

2. We need to construct a set of Targets- The 26×26 identity matrix would work for the initial sample: **eye(26)** is the Matlab command for the identity matrix (cute, huh?).
3. Use about 25 nodes in the hidden layer.

4. Use 75 percent of the data for training, 10 percent for validation, 15 percent for testing.
5. Initially, try using `trainlm`, then try `trainrp`. You'll find one is much much faster.
6. Construct your own testing error as a percentage of data misclassified by how much noise is present. Recall that the data is in blocks of 26 with increasing amounts of noise. For example, if X