I was never actually able to get my total error below 10^-5. I suspect this was a trick question because so much of my data overlapped with either side of my decision function. All four of my plots are on the same graph, so that it's easy to compare. I labeled them by color and there is a legend on the plot for them.

My data was normalized, so my mean height was 0.534377 and my mean weight was 0.391647
My standard deviation was

1 and 2 are best expressed through the code, but a simple explanation will be here. For the 75%% hard activation, it took me a while to understand the original implementation, but once I got it, the rest clicked into place. I started with initial weights of w1=0.125, w2 = -1, and w3 (bias) = 0.3. I chose a learning constant of 0.5 for my official data, although I did play around with the learning constant to see how it affected it. I noticed that as learning constant got smaller, the different iterations were making smaller and smaller changes to my weights, so I presume that a smaller learning constant would be most useful so that we aren't making wild jumps. This initial weight eventually changed for the 75% hard activation function to the weights of w1 = 1.49, w2 = 5.40, and w3 as 2.70.

25% Hard Activation came out to w1 = 1.03, w2 = -5.15, and w3 = 2.70

From a completely anecdotal observation, I noticed that the 75% data was closer to the line that I view as correct.

75% Soft Activation was w1 = 6.30, w2 = 23.67, w3 = 12.07
25% Soft Activation was w1 = 6.30, w2 = 23.02, w3 = 12.34

I noticed the difference for the 75% and the 25% was much smaller for the soft activation function. I believe this was do to the fact that our activation was no longer a straight function.

The largest difference I can see between hard and soft activation is the change in whether or not the neuron fires at certain values. By changing to a soft activation function, it gives a less rigid comparison against the threshold.