parallel

the method above is better than the serial method at training your

overall thetahat for all MF's... see the agregate results here

thetahat % prints thetahat

But it doesn't seem to fit each data point as well to the optimal, seen

in the plot of the actual output

Maybe this makes sense.... it's trying to fit \*all\* the data points to

the line at once, which won't result in a perfect fit for \*all\* of them

The serial method only has to fit one data point, which it does

perfectly, but then it can't model the overall best center points for all

the data with only one point, which leads to crappy results for training

the center points....wierd

serial

the method above does a good job of estimating thetahat for each

individual input, \*but\*, it's not necessarily good at training your

overall thetahat for all MF's... see the agregate results here

thetahat % prints thetahat

If we re-ran the normal Fuzzy PIE with these "trained" output center points,

you wouldn't get the same output results - which makes sense the way

that thetahat is constantly added to itself.

If you want to see the timeline, modify thetahat to be a matrix and

change the last line in the nested loop above to

thetahat(:,i) = thetahat(:,i) + step\_size \* cost\_gradient(:,i);









