n = 20; % number of points

points = [random('unid', 100, n, 1), random('unid', 100, n, 1)];

len = zeros(1, n - 1);

points = sortrows(points);

%% Initial set of points

plot(points(:,1),points(:,2));

for i = 1: n-1

len(i) = points(i + 1, 1) - points(i, 1);

end

while(max(len) > 2 \* min(len))

[d, i] = max(len);

k = on\_margin(points, i, d, -1);

m = on\_margin(points, i + 1, d, 1);

xm = 0; ym = 0;

%% New point

if(i == 1 || i + 1 == n)

xm = mean(points([i,i+1],1))

ym = mean(points([i,i+1],2))

else

[xm, ym] = dlg1(points([k, i, i + 1, m], 1), ...

points([k, i, i + 1, m], 2))

end

points = [ points(1:i, :); [xm, ym]; points(i + 1:end, :)];

end

%{

This is a block comment. Please ignore me.

%}

function [net] = get\_fit\_network(inputs, targets)

% Create Network

numHiddenNeurons = 20; % Adjust as desired

net = newfit(inputs,targets,numHiddenNeurons);

net.trainParam.goal = 0.01;

net.trainParam.epochs = 1000;

% Train and Apply Network

[net,tr] = train(net,inputs,targets);

end

foo\_matrix = [1, 2, 3; 4, 5, 6]''';

foo\_cell = {1, 2, 3; 4, 5, 6}''.'.';

cell2flatten = {1,2,3,4,5};

flattenedcell = cat(1, cell2flatten{:});