```
 classdef logical_object

    properties
        alpha=0;
        epoch=0;
        feature=[];
        target=[];
    end

    methods

        function self = logical_object(feature,target,epoch)

            self.feature = feature;
            self.target = target;

            if (isnumeric(epoch)) && (epoch~=0)
                self.epoch=epoch;
            else
                self.epoch=1500; %Default to 1500
            end
        end
        function [final_theta, err] = newton_vectorized(obj)

            period = obj.epoch/10;
            [m,n] = size(obj.feature);
            new_features = [ones(m,1), obj.feature];
            theta = zeros(obj.epoch,n+1);
            err = zeros(obj.epoch,1);

            for i = 1:obj.epoch

                hyp =
 logical_object.sigmoid(theta(i,:)*new_features')';

                err(i) = (1/m).*sum(-obj.target.*log(hyp)...
                        -(1-obj.target).*log(1-hyp));

                grad = (1/m).*new_features'*(hyp-obj.target);

%                 hess = (1/m).*sum(hyp.*(1-
hyp)).*new_features'*new_features;
                hess = (1/m).*new_features' * diag(hyp) * diag(1-hyp)
 * new_features;

                theta(i+1,:) = theta(i,:)-(inv(hess)*grad)';


                if (mod(i,period) == 0)
                fprintf('Epoch %2d: current theta values: %.2d, %.2d,
 \n %.2d. \n',...
                    i,theta(i,:));
                end
           end

        final_theta = theta(end,:);

    end
        function plot_graph(obj,theta,err)

%             fontSize = 20;

            subplot(1,2,1)

            pos = find(obj.target==1);
            neg = find(obj.target==0);

            plot(obj.feature(pos,1),obj.feature(pos,2),'r+'); hold on
            plot(obj.feature(neg,1),obj.feature(neg,2),'bo')
            line([0,100],[-theta(1)/theta(3),(-
(100*theta(2)+theta(1))/theta(3))])

 xlim([min(obj.feature(:,1)),max(obj.feature(:,1))]);ylim([min(obj.feature(:,2)),m
            xlabel('Exam 1 Score'); ylabel('Exam 2
 Score'); legend('Admitted','Not Admitted','Decision
 Boundary','location','best');


            subplot(1,2,2)
            plot(1:1:obj.epoch,err,'b')
            xlabel('epoch')
            ylabel('J_{\theta}')
            legend('Cost','location','best')

        end

    end

    methods(Static)
        function hyp = sigmoid(z)
            hyp = 1.0./(1.0+exp(-z));
        end
    end

end
```

```
Not enough input arguments.

Error in logical_object (line 14)
            self.feature = feature;
```

Published with MATLAB® R2016a