
```

classdef linear_regressor
    %{
    Args:
        alpha: 'int' setting the learning rate
        epoch: 'int' setting the number of epochs
        batch_size: 'int' setting the min-batch size

    Returns:
        new_theta: matrix of theta values
    %}
    properties
        alpha=0;
        epoch=0;
        feature=[];
        target=[];
    end

    methods

        function self = regressor_object(feature,target,alpha,epoch)

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

            if (isnumeric(alpha)) && (alpha~=0)
                self.alpha=alpha;
            else
                self.alpha=0.05; %Default to 0.05
            end
            if (isnumeric(epoch)) && (epoch~=0)
                self.epoch=epoch;
            else
                self.epoch=1500; %Default to 1500
            end
        end

        function [final_theta, err] = linear_regressor(obj,batch_size)

            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

                for j = 1:(n+1)

                    theta(i+1,j) = theta(i,j)-(obj.alpha*(1/batch_size)...
                        *sum(((sum((theta(i,:).*new_features),2)-
obj.target).*new_features(:,j)))));

```

```

        end

        if (mod(i,period) == 0)
            fprintf('Epoch %2d: current theta values %.2d and
%.2d. \n',...
                i,theta(end,1),theta(end,2));

        end

    end

    final_theta = theta(end,:);

end

function plot_graph(obj,theta)

    %         fontSize = 20;

    %         subplot(1,1,2)
    if length(theta) == 3

        plot3(obj.feature(:,1),obj.feature(:,end),...

            ((obj.feature(:,2)*theta(end,2))+(obj.feature(:,end)*theta(end,end))+theta(end,1)
            hold on

        scatter3(obj.feature(:,1),obj.feature(:,end),obj.target,'r')
            xlabel('feature_1')
            ylabel('feature_2')
            zlabel('target')
            legend('Regression','Training Data','location','best')
            hold off

    elseif length(theta) == 2

        plot(obj.feature,((obj.feature.*theta(end,end))...
            +theta(end,1)), 'k')
        hold on
        scatter(obj.feature,obj.target,'r')
        xlabel('feature_1')
        ylabel('target')
        legend('Regression','Training Data','location','best')
        hold off

    else

    end

end

function hyp_i = hypothesis(theta,features)

    hyp_i = sum((theta.*features),2)

```

```
end

end

end

Not enough input arguments.

Error in linear_regressor (line 39)
    period = obj.epoch/10;

Published with MATLAB® R2016a
```